(19) BUNDESREPUBLIK DEUTSCHLAND

(12) Übersetzung der europäischen Patentschrift

(51) Int. Cl.[5]: G 09 G 5/14
G 09 G 1/00

(87) EP 0 439 087 B1

(10) DE 691 23 465 T 2

DEUTSCHES PATENTAMT

(21) Deutsches Aktenzeichen: 691 23 465.5
(86) Europäisches Aktenzeichen: 91 100 660.9
(86) Europäischer Anmeldetag: 21. 1. 91
(87) Erstveröffentlichung durch das EPA: 31. 7. 91
(87) Veröffentlichungstag der Patenterteilung beim EPA: 11. 12. 96
(47) Veröffentlichungstag im Patentblatt: 26. 6. 97

(30) Unionspriorität: (32) (33) (31)
25.01.90 US 470728

(73) Patentinhaber:
Radius Inc., San Jose, Calif., US

(74) Vertreter:
BOEHMERT & BOEHMERT, 80801 München

(72) Erfinder:
Moss, Nicolas N., Sunnyvale, California 94086, US;
Marianetti II, Ronald, Palo Alto, California 94306, US

(84) Benannte Vertragstaaten:
AT, BE, CH, DE, DK, ES, FR, GB, GR, IT, LI, LU, NL, SE

(54) Verfahren zur Änderung der Abmessungen von Computeranzeigefenstern und ihrer Bewegung

DE 691 23 465 T 2

DE 691 23 465 T 2

RADIUS INC.
691 23 465.5-08
RL 2358

Computeranzeigebildschirme wurden traditionell so konzipiert, daß sie entweder eine "Porträt"-Orientierung vorsahen, bei der die vertikale Abmessung größer ist als die horizontale Abmessung, oder eine "Landschafts"-Orientierung, bei der die horizontale Abmessung größer ist als die vertikale Abmessung. Jüngste Fortschritte der Technologie machten es möglich, sowohl die Porträt-Orientierung als auch die Landschafts-Orientierung mit demselben Computer zu verwenden. Solche Computeranzeigesysteme erlauben dem Benutzer des Computers, einen (Anzeige-)Bildschirm zwischen der Porträt- und der Landschafts-Orientierung umzustellen, und einige Computer erlauben sogar die gleichzeitige Verwendung von mehreren Bildschirmen.

Wenn die Orientierung eines Bildschirms zwischen der Porträt- und der Landschafts-Orientierung neu eingestellt wird, verliert man zwangsläufig einen Teil des Anzeigebereichs aus der Sicht, und ein anderer Teil kommt hinzu. Bei einem Computersystem, welches "Fenster"-gestütze Anzeigen verwendet, kann der Bereich, welcher verloren geht, wichtige Teile eines Fensters (Windows) enthalten, die notwendig sind, damit der Benutzer das Fenster bewegen oder dessen Größe neueinstellen kann. Ferner kann es auch andere Bildschirme geben, welche Daten anzeigen, die ihre Orientierung nicht ändern. Um zu gewährleisten, daß die resultierende Anzeige nutzbar ist, wäre ein Verfahren zum automatischen Neudefinieren des Koordinatensystems der Computeranzeige in dem System sowie zum Bewegen von Fenstern und Neueinstellen ihrer Größe wünschenswert.

Teilen von zwei Fenstern;

Fig. 5 zeigt, wie der Bildschirm von Fig. 4 bei der vorliegenden Erfindung in der Landschafts-Orientierung aussieht;

Fig. 6a und 6b zeigen den Teil einer Anzeige auf einem Bildschirm mit fester Orientierung, sowohl bevor als auch nachdem die Orientierung eines zweiten zugeordneten Bildschirms von einer Porträt- in eine Landschafts-Orientierung umgestellt wurde;

Fig. 7 zeigt einen Bildschirm in Porträt-Orientierung mit einem Fenster;

Fig. 8 zeigt einen Bildschirm in Porträt-Orientierung mit demselben Fenster wie in Fig. 7, nachdem dieses Fenster auf herkömmliche Weise durch Zoomen vergrößert wurde;

Fig. 9 zeigt, wie der Bildschirm von Fig. 8 bei der vorliegenden Erfindung in Landschafts-Orientierung aussieht;

Fig. 10 zeigt einen Bildschirm in Landschafts-Orientierung mit demselben Fenster wie in Fig. 8, nachdem der Bildschirm von der Porträt- in die Landschafts-Orientierung umgestellt wurde, wobei die automatische Zoomfunktion gemäß der vorliegenden Erfindung aktiv ist;

Fig. 11 zeigt ein Steuerfeld-Anzeigefenster zum Auswählen von Merkmalen gemäß der vorliegenden Erfindung;

Fig. 11a und 11b zeigen jeweils im einzelnen den Teil des Steuerfeld-Anzeigefensters für die Neueinstellung der Fenstergröße, wobei die automatische Neuzoom-Funktion (Auto-Rezoom) der Fenstergrößenänderungs-Funktion aktiviert bzw. deaktiviert ist;

Fig. 12 zeigt ein Steuerfeld-Anzeigefenster für das Deaktivieren von Funktionen (Merkmalen) und Vorsehen von

benutzerspezifischen Befehlen für ausgewählte Fenstertypen gemäß der vorliegenden Erfindung;

Fig. 13a zeigt ein Blockdiagramm erfindungsgemäßer Software-Korrekturroutinen (Patches) für Softwareroutinen nach dem Stand der Technik;

Fig. 13b zeigt ein Blockdiagramm für Software-Unterstützungsroutinen gemäß der vorliegenden Erfindung;

Fig. 14 zeigt ein Flußdiagramm der erfindungsgemäßen Software-Korrekturroutine für die Routine _SlotManager nach dem Stande der Technik;

Fig. 15 zeigt ein Flußdiagramm der erfindungsgemäßen Software-Korrekturroutine für die Routine _GetNextEvent nach dem Stande der Technik;

Fig. 16 zeigt ein Flußdiagramm der erfindungsgemäßen Software-Korrekturroutine für die Routine _InitGraf nach dem Stande der Technik;

Fig. 17 zeigt ein Flußdiagramm der erfindungsgemäßen Software-Korrekturroutine für die Routine _NewWindow nach dem Stande der Technik;

Fig. 18 zeigt ein Flußdiagramm der erfindungsgemäßen Software-Korrekturroutine für die Routine _InitWindows nach dem Stande der Technik;

Fig. 19 zeigt ein Flußdiagramm der erfindungsgemäßen Software-Korrekturroutine für die Routine _SelectWindow nach dem Stande der Technik;

Fig. 20 zeigt ein Flußdiagramm der erfindungsgemäßen Software-Korrekturroutine für die Routine _CloseWindow nach dem Stande der Technik;

Fig. 21 zeigt ein Flußdiagramm der erfindungsgemäßen Software-Korrekturroutine für die Routine _DragWindow nach dem Stande der Technik;

Fig. 22 zeigt ein Flußdiagramm der erfindungsgemäßen Software-Korrekturroutine für die Routine _ShowHide nach dem Stande der Technik;

Fig. 23 zeigt ein Flußdiagramm der erfindungsgemäßen Soft-

ware-Korrekturroutine für die Routine _GrowWindow nach dem Stande der Technik;

Fig. 24 zeigt ein Flußdiagramm der erfindungsgemäßen Software-Korrekturroutine für die Routine _SizeWindow nach dem Stande der Technik;

Fig. 25 zeigt ein Flußdiagramm der erfindungsgemäßen Software-Korrekturroutine für die Routine _ZoomWindow nach dem Stande der Technik;

Fig. 26 zeigt ein Flußdiagramm der erfindungsgemäßen Software-Korrekturroutine für die Routine _MenuSelect nach dem Stande der Technik;

Fig. 27 zeigt ein Flußdiagramm der erfindungsgemäßen Software-Korrekturroutine für die Routine _TrackBox nach dem Stande der Technik;

Fig. 28 zeigt ein Flußdiagramm der erfindungsgemäßen Software-Korrekturroutine für die Routine _GetMouse nach dem Stande der Technik;

Fig. 29 zeigt ein Flußdiagramm der erfindungsgemäßen Software-Korrekturroutine für die Routine _Button nach dem Stande der Technik;

Fig. 30 zeigt ein Flußdiagramm der erfindungsgemäßen Software-Korrekturroutine für die Routine _TextBox nach dem Stande der Technik;

Fig. 31 zeigt ein Flußdiagramm der erfindungsgemäßen Software-Korrekturroutine für die Routine _InitZone nach dem Stande der Technik;

Fig. 32 zeigt ein Flußdiagramm der Softwareroutine Check/-Handle Flip gemäß der vorliegenden Erfindung;

Fig. 33a, 33b, 33c und 33d zeigen ein Flußdiagramm der Softwareroutine Compute New Window gemäß der vorliegenden Erfindung;

Fig. 34a und 34b zeigen ein Flußdiagramm der Softwareroutine Check Resize List gemäß der vorliegenden Erfindung;

Fig. 35 zeigt ein Flußdiagramm der Softwareroutine Resize

Window gemäß der vorliegenden Erfindung;

Fig. 36 zeigt ein Flußdiagramm der Softwareroutine Compute Resize Amqunt gemäß der vorliegenden Erfindung;

Fig. 37a, 37b, 37c und 37d zeigen ein Flußdiagramm der Softwareroutine Rebuild Desktop gemäß der vorliegenden Erfindung;

Fig. 38 zeigt ein Flußdiagramm der Softwareroutine Finder Cleanup gemäß der vorliegenden Erfindung;

Fig. 39 zeigt ein Flußdiagramm der Softwareroutine Map Rectangle to Main Device gemäß der vorliegenden Erfindung;

Fig. 40 zeigt ein Flußdiagramm der Softwareroutine Process a New Window gemäß der vorliegenden Erfindung;

Fig. 41a und 41b zeigen ein Flußdiagramm der Softwareroutine Get Window's Graphic Device gemäß der vorliegenden Erfindung.

In Fig. 1a ist ein Bildschirm eines Computers gezeigt, der entweder in Porträt-Orientierung 1 oder in Landschafts-Orientierung 2 betrieben werden kann. Bei einigen Bildschirmen ist es möglich, die Orientierung des Bildschirms zu jeder Zeit umzustellen oder zu kippen (flip). Fig. 1b zeigt, daß dann, wenn von der Porträt-Orientierung 1 in die Landschafts-Orientierung 2 gekippt wird, ein Teil des Anzeigebereiches 4 wegfällt, und ein entsprechender Teil 5 kommt hinzu, wobei ein Teil des Anzeigebereiches 3 weder wegfällt noch hinzukommt.

Fig. 2 zeigt eine Anzeige 1 in Porträt-Orientierung mit einem kleineren Fenster 10, einem größeren Fenster 11, einem Hauptmenübalken 12 und zwei Piktogrammen 13. Gemäß der vorliegenden Erfindung führt ein Kippen (flip) von der Porträt-Orientierung in Fig. 2 zu einer Veränderung der Größe oder Bewegung des kleineren und des größeren Fensters 10, 11, des Hauptmenübal-

kens 12 und der beiden Piktogramme 13. Die Anzeige, welche sich bei dem Kippen ergibt, ist in Fig. 3 gezeigt. Das kleinere Fenster 10 wird vertikal nach oben bewegt, damit es vollständig in die Grenzen der Landschaftsanzeige 2 paßt, die vertikale Abmessung des größeren Fensters 11 wird reduziert, damit es vollständig in die Grenzen der Landschaftsanzeige 2 paßt, die horizontale Abmessung des Hauptmenübalkens 12 wird vergrößert, damit dieser vollständig von der linken Seite der Landschaftsanzeige 2 zu der rechten Seite der Landschaftsanzeige 2 reicht, und die Piktogramme 13 werden von dem größeren Fenster 11 weiter weg bewegt.

Fenster und Menütext, welche vor einer Kippbewegung sichtbar und zugänglich waren, sollen vorzugsweise nach dem Kippen sichtbar und zugänglich bleiben. Die Teile des Fensters, welche vor dem Kippen zugänglich waren, sollen vorzugsweise auch nach dem Kippen zugänglich bleiben. Wenn das Anzeigesystem des Computers weitere Bildschirme außer dem, welcher gekippt wurde, verwendet, sollte die Auswirkung einer Kippbewegung auf diese anderen Bildschirme vorzugsweise minimal sein. Wenn der Hauptmenübalken vor dem Kippen sichtbar ist, muß er zusätzlich seine Größe abhängig von den neuen Abmessungen des gekippten Bildschirms ändern. Ähnlich müssen die Piktogramme bei beiden Orientierungen des Bildschirmes zugänglich bleiben.

Die vorliegenden Erfindung ermittelt daher, welcher Bildschirm in einem Computersystem mit mehreren Bildschirmen ein bestimmtes Fenster steuern soll. Dies wird erreicht, indem die Bedeutung der verschiedenen Ränder eines Fensters gewichtet werden, indem ermittelt wird, welcher Fensterrand auf welchem Bildschirm erscheint, und indem eine gewichtete Summe für jeden Bildschirm berechnet wird. Eine Ausführungsform der vorliegenden Erfindung verwendet das folgende Wichtungsverfahren:

8 wird zu der gewichteten Summe für einen Bildschirm addiert,

wenn dieser Bildschirm den oberen Rand des Fensters enthält.

4 wird zu der gewichteten Summe für einen Bildschirms addiert, wenn dieser Bildschirm den linken Rand des Fensters enthält.

2 wird zu der gewichteten Summe für einen Bildschirm addiert, wenn dieser Bildschirm den rechten Rand des Fensters enthält.

1 wird zu der gewichteten Summe für einen Bildschirm addiert, wenn dieser Bildschirm den unteren Rand des Fensters enthält.

Wenn somit ein Bildschirm das gesamte Fenster enthält, ist die Summe 15, und wenn der Bildschirm keinen Teil des Fensters enthält, ist die Summe 0. Der Bildschirm, welcher die höchste Summe für ein Fenster aufweist, soll dieses Fenster steuern. Wenn kein Bildschirm eine Summe größer als 0 enthält, wird das Fenster von dem Bildschirm gesteuert, welcher den Hauptmenübalken enthält. Der obere Rand eines Bildes wird als der wichtigste Rand betrachtet, weil sie normalerweise den Titelbalken für das Fenster enthält, welcher für die manuelle Verschiebung, Zoomen und das Schließen des Fensters verwendet wird.

Wenn ein Bildschirm, welcher ein vollständiges Fenster enthält, gekippt wird, wird bei der vorliegenden Erfindung die Größe des Fensters neu eingestellt und das Fenster nach Bedarf bewegt, um zu gewährleisten, daß das Fenster in seiner Gesamtheit auf dem gekippten Bildschirm bleibt. Ein Beispiel hierfür ist in den Fig. 1 und 2 gezeigt, wo das kleine und das große Fenster 10 und 11 nach dem Kippen vollständig innerhalb der Landschaftsanzeige 2 bleiben. Eine vertikale Bewegung des kleineren Fensters 10 macht es möglich, daß dieses in die Landschaftsanzeige 2 paßt. Das größere Fenster 10 in der Anzeige 1 der Fig. 2 mit Porträt-Orientierung ist zu hoch, um in die Landschaftsanzeige 2 der Fig. 3 zu passen, selbst wenn das größere Fenster 10 in vertikaler Richtung so weit wie möglich nach oben bewegt wird.

Daher wird die Größe des größeren Fensters 10 auf eine kleinere vertikale Abmessung neu eingestellt, damit das Fenster 10 in die Landschaftsanzeige 2 der Fig. 3 paßt. Bei der bevorzugten Ausführungsform der Erfindung wird diese Größenneueinstellung mittels Software erreicht, welcher den Code simuliert, welcher zum manuellen Verändern der Größe eines Fensters von einer Maus (Mouse) erzeugt wird.

Bei der bevorzugten Ausführungsform werden die Anfangsposition und Größe der Fenster 10 und 11 in einem Speicher gespeichert, wenn der Bildschirm gekippt wird, so daß das Kippen des Bildschirms zurück in seine ursprüngliche Orientierung zu einer Anzeige führt, welche identisch mit der anfänglichen Anzeige ist. Wenn der Benutzer jedoch die Größe oder Position von Fenstern auf dem gekippten Bildschirm ändert, führt das Zurückkippen des Bildschirms in seine ursprüngliche Orientierung nicht zu einer Anzeige, welche identisch mit der anfänglichen Anzeige ist.

In Fig. 4 sind Fenster 21 und 22 gezeigt, welche nur teilweise in der Porträtanzeige 1 enthalten sind. Gemäß der vorliegenden Erfindung wird beim Kippen des Bildschirms die obere linke Ecke jedes Fensters 21, 22 bei derselben Distanz zu dem Rand des Bildschirms gehalten, über welchen sich das Fenster hinauserstreckt, wie in Fig. 5 gezeigt. In Fig. 4 erstreckt sich das Fenster 21 unter die Unterkante 23 der Portraitanzeige 1. Wenn der Bildschirm gekippt wird, wie in Fig. 5 gezeigt, behält die obere linke Ecke des Fenster 22 ihren Abstand von der rechten Kante 24 der Landschaftsanzeige 2. Die vertikale Abmessung des Fensters 22 wird ferner bei der Änderung der Orientierung von Fig. 4 zu Fig. 5 verkleinert, um den unteren Rand des Fensters 22 bei der unteren Grenze der Bildschirmanzeige 2 zu halten, wie in Fig. 5 gezeigt. Wie bei der zuvor beschriebenen Situation werden die Anfangsposition und -größe der Fenster in dem Speicher gespeichert, um sie zu verwenden, wenn der Bildschirm

in seine ursprüngliche Orientierung zurückgekippt wird, ohne daß der Benutzer irgendwelche Änderungen an der Größe oder Position der Fenster vorgenommen hat.

In Fig. 6a wird ein Koordinatensystem dazu verwendet, die Positionen auf den Bildschirmen 34, 35 zu definieren. In diesem Fall hat der Bildschirm 34 eine Porträt-Orientierung auf einem drehbaren oder kippbaren Bildschirm, und der Bildschirm 35 hat eine Porträt-Orientierung auf einem Bildschirm mit fester Ausrichtung. Eine Ausführungsform der vorliegenden Erfindung verwendet ein Koordinatensystem, dessen Ursprung 30 bei dem obersten linken angezeigten Pixel liegt und welches eine erste Zahl aufweist, welche die Anzahl der Pixel unter dem Ursprung bezeichnet, sowie eine zweite Zahl, welche die Anzahl der Pixel rechts vom Ursprung bezeichnet. Wenn somit, wie in Fig. 6a gezeigt, die Porträt-Orientierung des Bildschirms 34 865 Pixel hoch und 640 Pixel breit ist, hat das unterste rechte Pixel 32 die Koordinaten (863, 639). Ähnlich hat das oberste linke Pixel 31 des Bildschirms 35 mit fester Ausrichtung die Koordinaten (0, 640), und das unterste rechte Pixel 33 des Bildschirms 35 mit festen Koordinaten hat die Koordinaten (863, 1279). Ein Fenster 38 welches vollständig in dem Bildschirm 35 mit festen Koordinaten enthalten ist, wie das in Fig. 6a gezeigte, hat eine obere linke Ecke 36 mit den Koordinaten (100, 800), und eine untere rechte Ecke 37 mit den Koordinaten (700, 1100).

Wenn der Bildschirm 34 gekippt wird, ändert sich das Koordinatensystem. In Fig. 6b ist der Porträtbildschirm 34 von Fig. 6a gezeigt, nachdem er in die Landschafts-Orientierung 41 gekippt wurde. Der Ursprung 42 des Bildschirms mit Landschafts-Orientierung 41 behält die Koordinaten (0, 0), die untere rechte Ecke 44 des Bildschirms mit Landschafts-Orientierung 41 hat nun jedoch die Koordinaten (639, 863). Das oberste rechte Pixel 43 des Bildschirms mit fester Orientierung 35 hat nun die Koordinaten (0, 864), und das unterste rechte Pixel 45 des

Bildschirms 35 hat nun die Koordinaten (863, 1503). Die Koordinaten der obersten rechten Ecke 39 und der untersten rechten Ecke 40 des Fensters 38 in dem Bildschirm mit fester Porträt-Orientierung 35 werden zu (100, 1024) bzw. (700, 1324). Das lokale Koordinatensystem des Bildschirms mit fester Orientierung 35 ändert sich somit aufgrund des Kippens des Bildschirms von der Porträt-Orientierung 34 in die Landschaftsorientierung 41. Bei der bevorzugten Ausführungsform werden die Koordinaten eines Fensters 38, welches von einem Bildschirm mit fester Orientierung 35 kontrolliert wird, so eingestellt, daß die Position des Fensters 38 relativ zu dem obersten linken Pixel 43 des Bildschirms mit fester Orientierung 35 unverändert bleibt, nachdem der andere Bildschirm 34 gekippt wurde.

Die Fig. 7 bis 10 zeigen, wie Fenster durch "Zoomen" gemäß der vorliegenden Erfindung vergrößert werden können. In Fig. 7 enthält der Bildschirm mit Porträt-Orientierung 1 ein Fenster 51 vollständig, er enthält ferner einen Titelbalken 12 und Piktogramme 13. Fig. 8 zeigt das Fenster 51 von Fig. 7, das mittels eines herkömmlichen Zoomverfahrens, welches aus dem Stand der Technik bekannt ist, vergrößert wurde. Das gezoomte Fenster 52 in Fig. 8 wurde so groß gemacht, daß es den Bildschirm so vollständig wie möglich ausfüllt, ohne den Menübalken 12 oder die Piktogramme 13 zu überdecken. Fig. 9 zeigt das Ergebnis einer Kippbewegung des Bildschirms mit Porträt-Orientierung 1 aus Fig. 8 in einen Bildschirm mit Landschafts-Orientierung 2. Die vertikale Abmessung des Fensters 52 in Fig. 8 ist bei dem Fenster 53 des Bildschirms mit Landschafts-Orientierung der Fig. 9 verkleinert, damit das gesamte Fenster 53 innerhalb des Bildschirms mit Landschafts-Orientierung 2 der Fig. 9 bleibt. Die Piktogramme 13 werden ebenfalls bewegt, so daß sie in den am weitesten rechts gelegenen Bereich des Bildschirms mit Landschafts-Orientierung 2 der Fig. 9 passen. Die übliche Zoom-Verarbeitung des Fensters 52 in Fig. 8 führt somit nicht zu einer ausreichenden Vergrößerung des Fensters 53, um

den Bildschirm zu füllen, nachdem der Bildschirm in die Landschafts-Orientierung 2 gekippt wurde, wie in Fig. 9 gezeigt. Um das Fenster 53 zu vergrößern, um den Bildschirm mit Landschafts-Orientierung 2 der Fig.9 vollständig zu füllen, muß ein zweites herkömmliches Zoom-Verfahren eingesetzt werden, um das Fenster 54 zu strecken, wie in Fig. 10 gezeigt. Das Fenster 54 wird somit mit unterschiedlichen Formfaktoren für die Höhe und Breite vergrößert, um den Bildschirm im wesentlichen so vollständig wie möglich zu füllen, ohne den Menübalken 12 oder die Piktogramme 13 zu überdecken.

Bei der bevorzugten Ausführungsform der vorliegenden Erfindung erfolgt die Zoom-Verarbeitung eines Fensters automatisch, wenn der Bildschirm gekippt wird. Fig. 10 zeigt das Ergebnis, welches man erhält, wenn der Bildschirm von Fig. 8 aus der Porträt- in die Landschafts-Orientierung gekippt wird, wobei das Zoomen des Fensters 52 automatisch erfolgt. Bei der vorliegenden Erfindung behält die automatische Zoom-Verarbeitung denselben relativen Abstand zwischen den Rändern des Fensters und den entsprechenden Rändern des Bildschirms bei, so daß der Menübalken 12 und die Piktogramme 13 zugänglich bleiben. Es kann eine übliche Datenverarbeitung (z.B. das Programm "Cleanup Desktop", welches von Apple Computer, Inc. erhältlich ist) dazu verwendet werden, die Piktogramme 13 abhängig von dem Kippen zwischen den Orientierungen des Bildschirms zu bewegen. Die Größe einiger Fenster, insbesondere der als "Dialogbox" bekannten Fenster, kann mittels herkömmlicher Fenster-Verarbeitungsverfahren nicht verändert werden, und ihre Größe wird auch bei der vorliegenden Erfindung nach dem Kippen zwischen den Bildschirm-Orientierungen nicht verändert.

In Fig. 11 ist die Anzeige eines Steuerfeldes 66 gezeigt, welches einem Benutzer ermöglicht, bestimmte Merkmale zu sperren, welche bei der vorliegenden Erfindung vorgesehen sind. Das Bild eines Schaltfeld- oder Steuerfeldschalters 61 auf dieser Anzei-

ge 66 gibt an, daß die Fensterpositionierungsfunktion aktiviert ist. Ein anderes Bild eines Steuerfeldschalters 62 auf dieser Anzeige 66 gibt an, daß die Fenstergrößen-Neueinstellungsfunktion aktiviert ist. Ein drittes Bild eines Steuerfeldschalters 63 auf dieser Anzeige 66 gibt an, daß die Finder-Aufräumfunktion, welche Piktogramme verschiebt, aktiviert ist. Das Bild eines Stellelementes 64 auf dieser Anzeige zeigt die Einstellung der Anzeige nach dem Kippen des Bildschirms an, um geringe Unregelmäßigkeiten auszugleichen. Ein weiteres Bild eines Schalters 65 ist auf der Anzeige 66 vorgesehen, welches anzeigt, daß alle Kippfunktionen aktiviert sind. Ein Benutzer kann jedes der Schalterbilder 61 bis 65 mittels der herkömmlichen Fenster-Tasten-Technologie verändern, indem ein Anzeigecursor (nicht gezeigt) über die gewünschte Schalterposition gelegt wird und eine Taste an der Maus des Computersystems (nicht gezeigt) angeklickt wird.

In den Fig. 11a und 11b sind weitere Einzelheiten in Bezug auf das Bild des Steuerfeldschalters 62 für die Veränderung oder Neueinstellung der Fenstergröße gezeigt. Das Bild dieses Steuerfeldschalters 62 zeigt zwei Funktionen. Die erste Funktion, welche bereits beschrieben wurde, besteht darin, daß die Fenstergrößenänderungsfunktion deaktiviert wird. Die zweite Funktion besteht darin, die automatische Zoomeinstellung (Auto-Rezoom-Funktion) gemäß der vorliegenden Erfindung zu deaktivieren. Wenn ein Benutzer mit der Maus zu dem Piktogramm 90 der Fig. 11a zeigt und dieses mit der herkömmlichen Fenster-Tasten-Technologie auswählt, ändert sich das Piktogramm 90 in das Piktogramm 91 der Fig. 11b, wodurch angezeigt wird, daß die Auto-Rezoom-Funktion deaktiviert wurde.

Eine neue Positionierung der Fenster, welche von einigen herkömmlichen Computerprogrammen vorgesehen werden, ist nicht machbar. Fig. 12 zeigt ein Steuerfeldfenster, welches einem Benutzer angibt, wie er bestimmte Fenster bestimmter Computer-

programme auf seine Bedürfnisse zuschneiden kann.

Die bevorzugte Ausführungsform dieser Erfindung nutzt mehrere Routinen des Standes der Technik zum Bewegen der Fenster und zum Verändern der Fenstergröße. Fig. 13a zeigt diese Routinen. Damit bei der Erfindung ein Fenster bewegt oder seine Größe verändert (Resize) werden kann, müssen für jedes Fenster, welches momentan in Gebrauch ist, bestimmte Informationen gespeichert werden. Die Routine NewWindow (neues Fenster) 104 des Standes der Technik realisiert die Zuordnung und Initialisierung der erforderlichen zusätzlichen Information. Wenn ein Fenster geschlossen ist, wird die zusätzliche Funktion nicht mehr benötigt. Die Routine CloseWindow (Fenster schließen) 107 des Standes der Technik wird dazu verwendet, den durch NewWindow reservierten Speicherplatz freizugeben. Die Routine GetNextEvent (nächstes Ereignis holen) 102 des Standes der Technik wird dazu verwendet, die auf das Kippen (Flip) bezogenen Funktionen, wie die Erfassung des Kippens, die Bewegung des Fensters und die Veränderung der Fenstergröße, mit dem Betriebssystem des Computers zu koordinieren.

Wenn ein Kippen erfaßt wird, wird bei der vorliegenden Erfindung eine Liste der vorhandenen Fenster erzeugt. Während Softwareaufrufen zu der Routine GetNextEvent 102, werden die Routinen GetMouse (Maus holen) 115 und Button (Taste) 116 des Standes der Technik dazu verwendet, eine manuelle Betätigung der Maus zu simulieren. Die Routine ShowHide (Verstecktes Zeigen) 109 des Standes der Technik wird dazu verwendet, vorhandene Fenster, welche durch ein laufendes Anwendungsprogramm der Sicht des Benutzers gerade verborgen sind, zu bewegen und ihre Größe zu verändern.

Die Routinen DragWindow (Fenster ziehen) 108, GrowWindow (Fenster vergrößern) 110 und ZoomWindow (Fenster zoomen) 112 des Standes der Technik werden nach einer Kippbewegung gemäß der

vorliegenden Erfindung dazu verwendet, die Grenzen einer Konstanten zu korrigieren, welche aus dem Stand der Technik bekannt ist und "screenBits.bounds" genannt wird und die die entsprechenden Parameter für das Ziehen, die Größeneinstellung und das Zoomen der Fenster einstellt. Einige Anwendungsprogramme verwenden Routinen des Standes der Technik wie SizeWindow (Fenstergröße einstellen) 111 anstelle von ZoomWindow 112. Diese Anwendungsprogramme arbeiten mit den Routinen TrackBox (Kästchen nachführen) 114 und SizeWindow 111 des Standes der Technik.

Die Routinen InitGraf (Grafik initialisieren) 103 und InitWindows (Fenster initialisieren) 105 werden gemäß der vorliegenden Erfindung dazu verwendet, die Größe des Speichers in einem internen Anzeigepuffer zu ermitteln, welche für ein Anwendungsprogramm reserviert werden muß, soweit eine Anwendung einen solchen Puffer unterstützt, damit die Anwendung eine richtige Anzeige sowohl in der Porträt- als auch in der Landschafts-Orientierung vorsehen kann. Anwendungsprogramme reservieren manchmal einen solchen Puffer, basierend auf der Konstanten "screenBits.bounds", um die Datendarstellung und Erneuerung zu beschleunigen.

Die Routine MenuSelect (Menü auswählen) 113 des Standes der Technik wird bei der vorliegenden Erfindung dazu verwendet, die Piktogramme in der herkömmlichen piktogrammgestützten Benutzerschnittstelle, welche üblicherweise in Verbindung mit der Erfindung eingesetzt wird, neu zu positionieren. Ähnlich wird die Routine TextBox (Text-Kästchen) 117 des Standes der Technik dazu verwendet, Anwendungsnamen in dem Menübalken neu zu positionieren, wenn nach einem Kippen diese Anwendungen gestartet werden.

Die Routinen InitZone (Zone initialisieren) 118, GetNextEvent (nächstes Ereignis holen) 102 und SelectWindow (Fenster aus-

wählen) 106 des Standes der Technik werden bei der vorliegenden Erfindung dazu verwendet, mehrere Anwendungsprogrammfenster zu verarbeiten, die in einer "MultiFinder"-Umgebung aktiv sein können, welche üblicherweise in Computern angetroffen werden, die mit der vorliegenden Erfindung arbeiten. In der MultiFinder-Umgebung kann es mehrere Programme geben, welche laufen, ohne daß direkt auf ihr Fenster zugegriffen werden kann ("Hintergrund Anwendungen"). Die MultiFinder-Umgebung erlaubt normalerweise nur das Verändern der Größe einer einzigen Fenstergruppe, nämlich jene der "Vordergrundanwendung". Bei der vorliegenden Erfindung wird eine Tabelle der aktiven Anwendungsprogramme erstellt, und eine Neueinstellung der Fenstergröße wird immer dann durchgeführt, wenn eine Anwendung als die Vordergrundanwendung ausgewählt wird. Die Routine InitGraf 103 des Standes der Technik wird dazu verwendet zu verhindern, daß Software-"Korrekturroutinen" (Patches) entfernt werden, welche in Übereinstimmung mit der vorliegenden Erfindung erstellt wurden, wenn in der MultiFinder Umgebung eine Neuanwendung gestartet wird.

Die Routine SlotManager (Steckplatz-Manager) 101 des Standes der Technik wird bei der vorliegenden Erfindung dazu verwendet, die Definitionen mehrerer Videogeräte für Betriebssystemversionen des Computers zu ermöglichen, der üblicherweise mit nur einem Videogerät, und somit einer Orientierung pro Hardware-Steckplatz arbeiten könnte. Bei der vorliegenden Erfindung wird jede mögliche Orientierung der Anzeige wie ein eigenes Gerät betrachtet, und es ist somit essentiell, daß mehrere Geräte zugelassen werden.

Ferner sind bei der bevorzugten Ausführungsform verschiedene Unterstützungs-Softwareroutinen 119 vorgesehen. Diese Routinen 119 sind im einzelnen in Fig. 13b gezeigt, und ihre zugehörigen Ablaufdiagramme sind in den Fig. 32 bis 41 angegeben.

Die in Fig. 13b gezeigte Routine Check/Handle Flipped Dolphin (Flipped Dolphin überprüfen/bearbeiten; abgekürzt "FlipOut") 3201 ist die Hauptroutine, um zu erfassen, wann ein Bildschirm gekippt wurde, und um die Fenster entsprechend zu verändern.

Die Routine Compute New Window Position (neue Fensterposition berechnen; abgekürzt "PositionWindow") 3301 berechnet und modifiziert die Grenzparameter eines Fensters abhängig von einem Kippen eines Bildschirmes.

Die Routine Check Resize List (Größenänderungsliste überprüfen) 3401 erstellt und verarbeitet die Liste der Fenster, deren Größe abhängig von einem Kippen eines Bildschirms verändert werden muß.

Die Routine Resize Window (Fenstergröße verändern) 3501 erzeugt virtuelle oder Phantom-Mausbetätigungen nach Bedarf, um die Größe eines Fensters abhängig von einem Kippen eines Bildschirms neu einzustellen.

Die Routine Compute Resize Amount (Umfang der Größenänderung berechnen; abgekürzt "CalcWindResize") 3601 berechnet den Umfang, in dem eine Fenstergröße abhängig von einem Kippen eines Bildschirms verändert werden muß, damit das Fenster zugänglich und auf der gekippten Anzeige logisch plaziert bleibt.

Die Routine Rebuild Desktop (Schreibtisch (Desktop) neu aufbauen) 3701 überprüft, ob mehrere Anzeigen in dem System verwendet werden, und wenn ja, ändert sie die Beziehungen zwischen den Bildschirmen abhängig von dem Kippen eines Bildschirms.

Die Routine Finder Cleanup (oder "CleanUpFinder"; Finder aufräumen) 3801 erzeugt virtuelle Maustastenbetätigungen abhängig von einem Kippen des Bildschirms, um eine Prozedur "Finder Cleanup" aus dem Stande der Technik auszulösen.

Die Routine Map Rectangle to Main Device (Rechteck auf Hauptgerät abbilden; abgekürzt "MapBigRect") 3901 bildet eine gegebene Fenstergröße ab, damit sie so gut wie möglich innerhalb die geltenden Abmessungen der Anzeige paßt, welche den Hauptmenübalken des Standes der Technik enthält.

Die Routine Process a New Window (neues Fenster verarbeiten; abgekürzt "ProcessNewWind") 4001 führt Zuordnungen durch, welche notwendig sind, um die Orientierungsänderungs- und Größenänderungs-Information auf dem aktuellen Stand zu halten, überprüft, ob die Fenstergröße in Übereinstimmung mit einer früheren Abmessung der Anzeige ist, zu der sie gehört, und erzeugt, falls dies nicht der Fall ist, virtuelle Maustastenbetätigungen, welche notwendig sind, um die Größe des Fensters auf die momentanen Abmessungen der zugehörigen Anzeige abzuändern.

Die Routine Get Window's Graphic Device (Grafikeinrichtung (GD) des Fensters holen; abgekürzt "GetWindGD") 4101 ermittelt, welche der mehreren Anzeigen einem Fenster am besten zugeordnet wird, wobei berücksichtigt wird, welche der oberen, linken, unteren und rechten Ränder des Fensters in jeder Anzeige enthalten sind.

Die Fig. 14 bis 31 zeigen mit weiteren Einzelheiten die Art, wie die in Fig. 13a gezeigten Routinen des Standes der Technik von der vorliegenden Erfindung "korrigiert" (oder "gepatcht") werden. Die Fig. 32 bis 41 zeigen mit weiteren Einzelheiten die Art, wie die zehn in Fig. 13b gezeigten Routinen gemäß der vorliegenden Erfindung arbeiten.

Wie man in Fig. 14 sieht, überprüft 1402 die Routine MySlotManager 1401, ob das System "32-Bit QuickDraw" des Standes der Technik läuft. Wenn ja, geht die Ausführung direkt zu dem _SlotManager-Code 1408 des Standes der Technik, weil die ver-

schiedenen Orientierungen der vorliegenden Erfindung über die "Videofamilien" von 32-Bit QuickDraw realisiert werden und somit keine Daten modifiziert werden müssen. Anderenfalls wird überprüft 1403, ob der aufrufende Algorithmus nach einer Kartendatenstruktur sucht, welche modifiziert werden sollte, um die momentane Orientierung gemäß der vorliegenden Erfindung wiederzugeben. Wenn der aufrufende Algorithmus nicht nach einer solchen Struktur sucht, springt die Ausführung unmittelbar zu dem ursprünglichen _SlotManager-Code 1408. Anderenfalls wird überprüft 1404, ob der aufrufende Algorithmus nach Videoschaltungskarten-Daten sucht, welche Parameter für einen Videomodus spezifizieren. Wenn nein, springt die Verarbeitung unmittelbar zu dem ursprünglichen _SlotManager-Code 1408. Anderenfalls wird überprüft 1405, ob der aufrufende Algorithmus nach Videoparametern-Kartendaten einer Karte (Leiterplatte) sucht, welche Änderungen in der Orientierung der Anzeige unterstützt. Wenn nein, springt die Ausführung unmittelbar zu dem ursprünglichen _SlotManager-Code 1408. Anderenfalls wird untersucht 1406, ob die Karte, für welche die Videoparameter angefordert werden, in der Landschaftsorientierung ist. Wenn nein, springt die Ausführung unmittelbar zu dem _SlotManager-Code 1408 des Standes der Technik. Anderenfalls wird das Landschafts-Bit der Identifikationsnummer der Karte, deren Videoparameter angefordert werden, gesetzt. Schließlich geht die Ausführung mit der modifizierten Identifikationsnummer zu dem _SlotManager-Code 1408 des Standes der Technik weiter. Diese Korrektur (Patch) ermöglicht einem Betriebssystem, welches nicht mit 32-Bit QuickDraw arbeitet, für beide Orientierungen gemäß der vorliegenden Erfindung und unter Verwendung nur einer Kartenidentifikationsnummer auf die Videoparameter zuzugreifen. Die Korrektur fängt Anfragen nach Videoparametern gemäß der vorliegenden Erfindung ab und modifiziert die Kartenidentifikationszahl, wenn die Karte im Landschaftsmodus ist. Der _SlotManager-Code des Standes der Technik gibt somit die Videoparameter der momentanen Orientierung gemäß der vorliegenden Erfindung zurück und simu-

liert so die Funktionen der "Videofamilien"-Konzepte des Standes der Technik, welche 32-Bit QuickDraw vorsieht.

Wie in Fig. 15 gezeigt, führt die Routine MyGetNextEvent 1501 einen Schritt 1502 durch, um die Liste der Fenster zu überprüfen, deren Größe nach einer Kippbewegung neu festgelegt werden muß. Sie führt dann den _GetNextEvent-Code 1503 des Standes der Technik aus. Danach wird ein Schritt 1504 ausgeführt, indem jedes Kippen, das aufgetreten ist, geprüft und verarbeitet wird. Schließlich führt sie einen Schritt 1505 aus, bei dem die erforderliche Filterung durchgeführt wird, um die Verarbeitung der virtuellen Maustastenbetätigungen, die von der vorliegenden Erfindung erzeugt werden, zu unterstützen. Diese Korrekturroutine sieht für die bevorzugte Ausführungsform der vorliegenden Erfindung eine passende Stelle zum Erfassen und Reagieren auf eine neue Monitororientierung vor, die sich ergibt, wenn der Benutzer eine Anzeige kippt. Sie koordiniert die Größenänderungsfunktionen, die erforderlich sind, um Fenster für die neue Anzeigeorientierung zu verändern, erfaßt Änderungen der Orientierung und sieht die während der Verarbeitung der virtuellen Maustastenbetätigungen erforderliche Filterung vor, um alle Effekte der vorliegenden Erfindung zu realisieren.

Bei der in Fig. 16 gezeigten Routine MyInitGraf 1601 wird geprüft 1602, ob auf der Maschine MultiFinder läuft, und wenn ja, werden die Schritte 1603 und 1604 ausgeführt, um die Korrekturroutinen für die Routinen _InitWindow und _CloseWindow des Standes der Technik neu zu installieren. Anderenfalls geht die Steuerung direkt zu der Ausführung des Schrittes 1605 weiter, der den _InitGraf-Code des Standes der Technik aufruft. Schließlich wird bei Bedarf im Schritt 1606 die globale Variable "screenBits.bounds" des Standes der Technik auf ein Rechteck eingestellt, welches die Dimensionen beider Orientierungen gemäß der vorliegenden Erfindung umfaßt. Diese Korrekturroutine tut zwei Dinge. Wenn das Programm MultiFinder des

Standes der Technik läuft, installiert sie erstens die Korrekturroutinen für die Routinen _InitWindows und _CloseWindows des Standes der Technik neu, welche entfernt worden sein können. Wenn sie von einer Anwendung aufgerufen wurde, welche von dem Benutzer als "inkompatibel" spezifiziert wurde, stellt sie zweitens die globale Variable "screenBits.bound" des Standes der Technik auf ein Rechteck ein, welches die Abmessungen beider Orientierungen gemäß der vorliegenden Erfindung umfaßt.

Die in Fig. 17 gezeigte Routine MyNewWindow 1701 ruft den _NewWindow-Code 1702 des Standes der Technik auf und führt dann den Schritt 1703 aus, der eine Datenstruktur zuordnet und initialisiert, welche von der vorliegenden Erfindung verwendet wird, um das Fenster nach einer Änderung der Anzeigeorientierung neu zu positionieren und/oder seine Größe zu verändern. Diese Korrekturroutine sieht eine passende Stelle für die vorliegende Erfindung zum Erzeugen zusätzlicher Datenstrukturen vor, welche zum Verarbeiten des neuen Fensters nach einer Änderung der Anzeigeorientierung notwendig sind.

Die in Fig. 18 gezeigte Routine FlipInitWindows 1801 führt einen Schritt 1802 aus, um die momentane untere rechte Ecke des globalen Parameters "screenBits.bounds" des Standes der Technik zurückzuholen. Der nächste Schritt 1803 wird ausgeführt, um die untere rechte Ecke der globalen Variable "screenBits.bounds" des Standes der Technik an die momentanen Abmessungen der Anzeige, welche den Menübalken enthält, anzupassen. Dann wird der _InitWindows-Code 1804 des Standes der Technik ausgeführt. Danach wird ein Schritt 1805 ausgeführt, um die untere rechte Ecke der globalen Variable "screenBits.bounds" auf ihren ursprünglichen Wert zurückzustellen. Als nächstes wird ein Schritt 1806 ausgeführt, um sicherzustellen, daß die Fenster, für welche die vorliegende Erfindung eine eigene Datenstruktur reserviert hat, in dem System noch existieren, und falls dies nicht der Fall ist, um diese zusätzliche Datenstruktur frei-

zugeben. Schließlich geht die Ausführung zu dem aufrufenden Programm zurück 1807. Diese Korrekturroutine tut zwei Dinge. Erstens gewährleistet sie, daß die untere rechte Ecke der Globalen "screenBits.bounds" des Standes der Technik während der Ausführung der _InitWindows-Routine des Standes der Technik richtig ist. Zweitens gewährleistet sie, daß alle Fenster, für welche die vorliegende Erfindung eine eigene Datenstruktur reserviert hat, noch existieren.

Die in Fig. 19 gezeigte Routine FlipSelectWindow 1901 führt einen Schritt 1902 durch, um dem Zeiger zu dem ausgewählten Fenster in einem globalen Speicher zu speichern. Dann geht die Ausführung direkt zu dem _SelectWindow-Code 1903 des Standes der Technik weiter. Diese Korrekturroutine aktualisiert den Zeiger zu dem obersten Fenster auf dem Schreibtisch (Desktop) gemäß der bevorzugten Ausführungsform, der benötigt wird, um in einer MultiFinder-Umgebung zu ermitteln, ob eine neue Anwendung zur aktuellen Anwendung geworden ist.

Die in Fig. 20 gezeigte Routine FlipCloseWindow 2001 führt einen Schritt 2002 durch, welcher die gemäß der vorliegenden Erfindung beim Erzeugen des Fensters reservierte zusätzliche Datenstrukturen freigibt. Schließlich geht die Ausführung direkt zu dem _CloseWindow-Code des Standes der Technik weiter. Diese Korrekturroutine gibt die zusätzliche Datenstruktur frei, welche beim Erzeugen des Fensters von der vorliegenden Erfindung erzeugt wurde, weil nach dem Schließen des Fensters die zusätzliche Datenstruktur nicht mehr gebraucht wird.

Die in Fig. 21 gezeigte Routine FlipDragWindow 2101 führt einen Schritt 2102 durch, der eine Kopie des Parameters "boundsRect" erzeugt. Dann wird ein Schritt 2103 ausgeführt, bei dem die "boundsRect"-Kopie neu abgebildet wird, um die Abmessungen der Anzeige, welche den Menübalken enthält, falls notwendig, anzupassen. Als nächstes wird der _DragWindow-Code 2104 des Standes

der Technik mit der modifizierten Kopie des "boundsRect"-Parameters ausgeführt. Dann wird ein Schritt 2105 ausgeführt, welcher die "boundsRect"-Kopie freigibt. Schließlich geht die Ausführung zu dem aufrufenden Programm zurück 2106. Diese Korrekturroutine gewährleistet, daß der Parameter "boundsRect" zu den Abmessungen der Anzeige paßt, welche den Menübalken enthält. Anwendungen rufen normalerweise die _DragWindow-Routine des Standes der Technik mit einem "boundsRect" auf, welches mit den Abmessungen der Anzeige übereinstimmt, die den Menübalken enthielt, als die Anwendung gestartet wurde. Wenn die Anzeige eine kippbare Anzeige ist und der Benutzer die Anzeige dann kippt, paßt der Parameter "boundsRect" für nachfolgende _DragWindow-Aufrufe nicht zu den aktuellen Abmessungen der Anzeige. Diese Korrekturroutine bildet dann eine Kopie des Parameters "boundsRect" neu ab, damit er zu der aktuellen Abmessung der gekippten Anzeige paßt.

Die in Fig. 22 gezeigte Routine FlipShowHide 2201 überprüft 2202, ob die Routine _ShowHide des Standes der Technik aus der vorliegenden Erfindung heraus aufgerufen wurde. Wenn ja, geht die Ausführung direkt zu dem _ShowHide-Code 2210 des Standes der Technik weiter. Anderenfalls wird überprüft 2203, ob die zusätzliche Datenstruktur existiert, die von der vorliegenden Erfindung normalerweise dem Fenster zugeordnet wird. Wenn keine solche Datenstruktur vorhanden ist, geht die Ausführung direkt zu dem _ShowHide-Code 2210 des Standes der Technik weiter. Anderenfalls wird ein Schritt 2204 ausgeführt, der den sichtbaren Status des Fensters in die zusätzliche Datenstruktur kopiert. Als nächstes wird überprüft 2205, ob die aufrufende Prozedur das Fenster verbergen möchte. Wenn das Fenster verborgen werden soll, geht die Ausführung direkt zu dem ShowHide-Code 2210 des Standes der Technik weiter. Anderenfalls wird überprüft, ob das Fenster sichtbar ist. Wenn dies der Fall ist, geht die Übertragung direkt zu dem ShowHide-Code 2210 des Standes der Technik weiter. Anderenfalls wird ein Schritt 2207

ausgeführt, um die aktuelle Position/den Ort des Fensters zu ermitteln. Dann wird ein Schritt 2208 ausgeführt, um die Position des Fensters (nach Bedarf) auf die aktuelle Schreibtischkonfiguration einzustellen, wenn der Benutzer dies erlaubt. Wenn das Fenster zu einem Typ gehört, dessen Größe verändert werden kann, wird ein Schritt 2209 ausgeführt, um das Fenster zu der Liste der Fenster zuzufügen, deren Größe verändert werden soll. Schließlich geht die Ausführung direkt zu dem _ShowHide-Code 2210 des Standes der Technik weiter. Diese Korrekturroutine erleichtert die Neupositionierung und neue Einstellung der Größe der Fenster, welche nicht sichtbar waren, als der Benutzer die Orientierung des Bildschirms gemäß der vorliegenden Erfindung gekippt hat.

Die in Fig. 23 gezeigte Routine FlipGrowWindow 2301 führt einen Schritt 2302 durch, welcher ein Rechteck reserviert, der oberen linken Ecke (4, Menübalkenhöhe+4) zuordnet und die untere rechte Ecke des Parameters sizeRect auf die momentane untere rechte Ecke des Rechtecks kopiert. Dann wird ein Schritt 2303 ausgeführt, um das momentane Rechteck auf die aktuelle Größe der Anzeige abzubilden, welche den Menübalken enthält. Danach wird die Routine _GrowWindow 2304 des Standes der Technik ausgeführt, wobei anstelle des Parameters sizeRect des Standes der Technik das momentane Rechteck verwendet wird. Schließlich gibt der Schritt 2305 das momentane Rechteck frei. Diese Korrekturroutine gewährleistet, daß der Parameter sizeRect, der an die _GrowWindow-Routine des Standes der Technik übergeben wird, zu der aktuellen Größe der Anzeige paßt, die den Menübalken enthält.

Die in Fig. 24 gezeigte Routine FlipSizeWindow 2401 führt einen Schritt 2402 aus, der ein temporäres Rechteck erzeugt. Dann wird überprüft 2403, ob die Routine _SizeWindow des Standes der Technik aufgerufen ist, um eine Fensterzoom-Funktion zu realisieren. Wenn dies nicht der Fall ist, geht die Ausführung

direkt zu dem _SizeWindow-Code 2408 des Standes der Technik weiter. Anderenfalls wird ein Schritt 2404 ausgeführt, um die Position und Größe des neuen Fensters in das temporäre Rechteck zu kopieren. Als nächstes wird ein Schritt 2405 ausgeführt, um das temporäre Rechteck auf die aktuelle Größe der Anzeige mit dem Menübalken abzubilden. Dann wird überprüft 2406, ob zusätzlich die Größe des Fensters verändert werden muß, so daß es auf die Anzeige mit dem Menübalken paßt. Wenn nein, geht die Ausführung direkt zu dem _SizeWindow-Code 2408 des Standes der Technik weiter. Anderenfalls erzeugt ein Schritt 2407 virtuelle Mausbetätigungen, indem der Code simuliert wird, welcher auf der Basis des neu abgebildeten temporären Rechtecks erzeugt wird, wenn ein Benutzer eine Maus führt und anklickt, um die Größe des Fensters weiter zu verändern. Dann wird der _SizeWindow-Code 2408 des Standes der Technik aufgerufen. Als nächstes wird ein Schritt 2409 ausgeführt, welcher die neue Position und Größe des Fensters speichert, wenn die Routine _SizeWindow des Standes der Technik aktiviert war. Schließlich wird ein Schritt 2410 ausgeführt, um das temporäre Rechteck freizugeben. Diese Korrekturroutine stellt sicher, daß Anwendungen, welche Fenster mit Hilfe der Routine _SizeWindow des Standes der Technik zoomen (anstelle der Routine _ZoomWindow des Standes der Technik), das Fenster richtig auf die aktuelle Größe der Anzeige, welche den Menübalken enthält, zoomen.

Die in Fig. 25 gezeigte Routine FlipZoomWindow 2501 überprüft 2502, ob das Fensterrechteck für den Standardzustand gleich dem Rechteck für einen Benutzerzustand ist. Wenn nein, geht die Steuerung direkt zum Schritt 2504 weiter. Anderenfalls wird ein Schritt 2503 aufgerufen, um das Rechteck des Benutzerzustands auf die aktuelle Größe der Anzeige, welche den Menübalken enthält, abzubilden. Als nächstes bildet der Schritt 2504 das Rechteck des Standardzustands auf die aktuelle Größe der Anzeige mit dem Menübalken ab. Danach wird der _ZoomWindow-Code 2505 des Standes der Technik ausgeführt. Als nächstes wird überprüft

2506, ob die Routine _ZoomWindow des Standes der Technik aufgerufen war, um ein Fenster neu zu zoomen, so daß es an die neue Orientierung der Anzeige nach einem Kippen der Anzeige durch den Benutzer angepaßt ist. Wenn dies nicht der Fall war, geht die Ausführung unmittelbar zu dem aufrufenden Programm zurück 2508. Wenn doch, stellt ein Schritt 2507 für das (Benutzerzustands)-Rechteck dessen Zustand vor dem Zoomen wieder her. Als nächstes geht die Ausführung zu dem aufrufenden Programm zurück 2508. Diese Korrekturroutine führt zwei Funktionen aus. Zunächst stellt sie sicher, daß die Benutzerzustands- und Standardzustands-Rechtecke für das Fenster, welches gezoomt wird, richtig auf die aktuellen Abmessungen der Anzeige mit dem Menübalken abgebildet werden. Die zweite Funktion, welche von der Korrekturroutine ausgeführt wird, erlaubt dem Benutzer, nach einem erneuten Zoomen zu einem kleinen Standardzustand zurückzuzoomen. Dies wird erreicht, indem das kleine Standardzustand-Rechteck gespeichert und sein Wert nach dem Aufruf der Routine _ZoomWindow des Standes der Technik wiederhergestellt wird (welche das Standardzustands-Rechteck auf einen unerwünschten Wert geändert haben könnte).

Die in Fig. 26 gezeigte Routine FlipMenuSelect 2601 überprüft 2602, ob der Finder aufgerufen ist, um eine Aufräumoperation (Cleanup) durchzuführen. Wenn nein, geht die Steuerung direkt dazu dem _MenuSelect-Code 2603 des Standes der Technik weiter. Anderenfalls wird der _MenuSelect-Code 2604 des Standes der Technik als ein Unterprogramm aufgerufen. Als nächstes ersetzt ein Schritt 2605 das Ergebnis des Aufrufs durch den Menüpunkt-Identifikator von Special-Cleanup. Schließlich geht die Steuerung zu dem aufrufenden Programm zurück 2606. Diese Korrekturroutine erleichtert den Aufruf einer Finder-Aufräumoperation, indem sie den Menüpunkt-Identifikator von Special-Cleanup zurückgibt, wenn sie aufgerufen wird.

Die in Fig. 27 gezeigte Routine FlipTractBox 2701 ruft den

_TrackBox-Code 2702 des Standes der Technik auf. Dann führt sie einen Schritt 2703 aus, der das Ergebnis in einen globalen Speicher kopiert. Schließlich kehrt sie zu dem aufrufenden Programm 2704 zurück. Diese Korrekturroutine speichert das Ergebnis des Aufrufs der _TrackBox-Routine des Standes der Technik, so daß sie später von der _SizeWindow-Korrekturroutine untersucht werden kann, um zu ermitteln, ob _SizeWindow aufgerufen wurde, um ein Fenster zu zoomen.

Die in Fig. 28 gezeigte Routine FlipGetMouse 2801 ruft den _GetMouse-Code 2802 des Standes der Technik auf. Dann überprüft sie 2803, ob eine virtuelle Mausbetätigung erzeugt wird. Wenn nein, geht die Steuerung sofort zu dem aufrufenden Programm 2807 zurück. Anderenfalls wird eine Überprüfung 2804 durchgeführt, um sicherzustellen, daß die virtuelle Mausbetätigung noch immer in der Ereignisschlange des Systems ist. Wenn ja, wird ein Schritt 2805 ausgeführt, der die virtuelle Mausposition anstelle der tatsächlichen Position der Maus auf der Anzeige zurückgibt. Wenn nein, wird ein Schritt 2806 ausgeführt, der die virtuelle Mausbetätigung löscht. In jedem Fall geht die Ausführung als nächstes zu dem aufrufenden Programm 2807 zurück. Diese Korrekturroutine ermöglicht, daß Anwendungsprogramme virtuelle Mausbetätigungen akzeptieren, welche gemäß der vorliegenden Erfindung erzeugt werden. Sie gibt die virtuelle Position der Maus anstelle der tatsächlichen Position zurück, wenn eine virtuelle Mausbetätigung erzeugt wurde.

Die in Fig. 29 gezeigte Routine FlipButton 2901 führt den _Button-Code 2902 des Standes der Technik aus. Als nächstes überprüft sie 2903, ob eine virtuelle Mausbetätigung erzeugt wurde. Wenn nein, geht die Steuerung sofort zu dem aufrufenden Programm 2907 zurück. Anderenfalls wird überprüft 2904, ob der Tastenzähler gemäß der vorliegenden Erfindung größer als null ist. Wenn nein, geht die Ausführung sofort zu dem aufrufenden Programm 2907 zurück. Anderenfalls wird ein Schritt 2905 aus-

geführt, der den Tastenzähler dekrementiert. Als nächstes wird ein Schritt 2906 aufgerufen, der den zurückgegebenen Wert der Routine _Button des Standes der Technik immer auf WAHR setzt. Schließlich geht die Ausführung zu dem aufrufenden Programm 2907 zurück. Diese Korrekturroutine erlaubt Anwendungsprogrammen, die virtuellen Mausereignisse anzunehmen, welche gemäß der vorliegenden Erfindung erzeugt werden. Sie gibt für eine spezielle Anzahl Iterationsschritte anstelle des tatsächlichen Zustands der Maustaste immer dann WAHR zurück, wenn die bevorzugte Ausführungsform eine virtuelle Maustastenbetätigung erzeugt.

Die in Fig. 30 gezeigte Routine FlipButton 3001 ruft einen Schritt 3002 auf, welcher ein temporäres Rechteck erzeugt. Dann überprüft sie 3003, ob der momentane Zeichenport zu der Routine WindowManager des Standes der Technik gehört. Wenn nein, geht die Steuerung zum Schritt 3006 weiter. Anderenfalls wird überprüft 3004, ob die Textbox in dem Menübalken liegt. Wenn nein, geht die Steuerung zum Schritt 3006 weiter. Anderenfalls wird ein Code 3005 ausgeführt, welcher dem temporären Rechteck die richtige Größe für die momentane Orientierung der Anzeige mit dem Menübalken zuordnet und die ursprünglichen Parameter des Kästchens durch das temporäre Rechteck ersetzt. Als nächstes wird der _TextBox-Code 3006 des Standes der Technik ausgeführt. Der nächste Schritt 3007 gibt das temporäre Rechteck wieder frei. Schließlich geht die Ausführung zu dem aufrufenden Programm 3008 zurück. Diese Korrekturroutine behebt eine Inkompatibilität mit Anzeigen, welche mehrere Orientierungen haben können, wie die gemäß der vorliegenden Erfindung, welche sich aufgrund des Finders ergibt. Sie gewährleistet, daß der Finder die Namen aufgerufener Anwendungen in die Mitte des Menübalkens setzt.

Die in Fig. 31 gezeigte Routine FlipInitZone 3101 führt einen Schritt 3102 aus, der alle Einträge aus der Prozeßtabelle gemäß

der bevorzugten Ausführungsform entfernt, welche einen ZonePtr (Zonenzeiger) haben, der innerhalb einer neuen Gruppenzone liegt, die initialisiert wird. Als nächstes geht die Steuerung zu dem _InitZone-Code 3103 des Standes der Technik weiter. Diese Korrekturroutine unterstützt die bevorzugte Ausführungsform bei der Aufrechterhaltung einer Prozeßtabelle, wenn sie in einer MultiFinder-Umgebung arbeitet. Es gibt kein allgemein akzeptiertes Verfahren zum Ermitteln, wann ein Prozeß in einer MultiFinder-Umgebung endet. Die Tabelle der Prozesse gemäß der bevorzugten Ausführungsform wird daher aktualisiert, wann ein neuer Prozeß gestartet wird, währenddessen seine Zone initialisiert wird.

Die in Fig. 32 gezeigte Routine FlipOut 3201 überprüft 3202, ob ein Monitor gekippt worden ist. Wenn nein, geht die Steuerung zu dem aufrufenden Programm 3213 zurück. Anderenfalls wird ein Schritt 3203 ausgeführt, bei dem das Bild auf dem gekippten Bildschirm ausgeblendet wird, bis es schwarz ist. Danach wird ein Schritt 3204 ausgeführt, der Fenster überprüft, um zu ermitteln, ob sie in ihrem Standardzustand (oder "Zoomed-Out") sind. Als nächstes wird ein Schritt 3205 ausgeführt, der eine Nachbarschaftsabbildung aller Anzeigen aufbaut, die momentan in dem System arbeiten. Dann wird ein Schritt 3206 ausgeführt, um für die gekippte Anzeige die Größe der globalen GDevice-Daten des Standes der Technik neu festzulegen. Dann wird ein Schritt 3207 ausgeführt, um nach Bedarf die aktiven Anzeigen neu zu positionieren (um der Änderung der Abmessungen des gekippten Gerätes zu begegnen), und um das System-Betriebsmittel "Scrn" (Screen = Bildschirm) des Standes der Technik zu aktualisieren. Dann wird ein Schritt 3208 ausgeführt, um alle GrafPorts (Grafikports) des Standes der Technik zu modifizieren, welche auf dem gekippten Gerät basieren, und diese für die neue Orientierung des Gerätes zu aktualisieren. Dann wird ein Schritt 3209 ausgeführt, um die Globale GrayRgn des Standes der Technik neu aufzubauen, damit sie mit der neuen Konfiguration der Anzeige

übereinstimmt. Dann wird ein Schritt 3210 ausgeführt, um Fenster nach Bedarf neu zu positionieren, so daß sie auch bei Neukonfiguration der Anzeigen noch zugänglich sind. Dann wird ein Schritt 3211 ausgeführt, um die Maus/Cursor-Datenstrukturen des Standes der Technik neu aufzubauen. Dann wird ein Schritt 3212 ausgeführt, um die Anzeigen neu darzustellen und das Bild auf der gekippten Anzeige wieder einzublenden, so daß es wieder sichtbar ist. Schließlich geht die Steuerung zu dem aufrufenden Programm 3213 zurück. Dieser Code ermittelt, ob eine Anzeige gekippt wurde, und wenn dies der Fall ist, sieht er die gesamte Grundverarbeitung vor, um die globalen Strukturen des Betriebssystems des Standes der Technik zu aktualisieren, um der Änderung möglichst gut zu begegnen.

Die in den Fig. 33a, 33b, 33c und 33d gezeigte Routine Position Window 3301 überprüft 3302, ob das Fenster dieselbe Größe hat wie eine der möglichen Orientierungen der Anzeige, welche den Menübalken enthält. Wenn ja, geht die Steuerung direkt zum Schritt 3326 weiter. Wenn nein, wird ein Schritt 3303 ausgeführt, der den minimalen oberen linken Wert berechnet, welche die Variable portRect des Standes der Technik für das Fenster annehmen kann. Dann wird ein Schritt 3304 aufgerufen, welcher das eingrenzende Rechteck der Anzeige zurückgibt, welcher das Fenster am logischsten zugeordnet wird. Dann wird überprüft 3305, ob die Anzeige, der das Fenster zugeordnet ist, die Anzeige mit dem Menübalken ist. Wenn nein, geht die Steuerung zum Schritt 3307 weiter. Anderenfalls wird ein Schritt 3306 ausgeführt, der das eingrenzende Rechteck der zugehörigen Anzeige modifiziert, so daß es die minimale obere und untere Versetzung wiedergibt, welche das Fenster innerhalb seiner zugehörigen Anzeige einhalten muß (welche von dem Benutzer anwendungsweise bestimmt wird; die voreingestellten Werte sind 0). Dann wird überprüft 3307, ob die zugehörige Anzeige den unteren Rand des Fensters enthielt, bevor der Benutzer die Anzeige gekippt hat. Wenn nein, geht die Steuerung zum Schritt 3310 weiter. Anderen-

falls wird überprüft 3308, ob die zugehörige Anzeige noch immer den unteren Rand des Fensters enthält. Wenn ja, geht die Steuerung zum Schritt 3310 weiter. Anderenfalls wird ein Schritt 3309 ausgeführt, der das Fenster nach oben bewegt, so daß der untere Rand auf der zugehörigen Anzeige bleibt. Dann wird überprüft 3310, ob die zugehörige Anzeige den rechten Rand des Fensters enthielt, bevor der Benutzer die vorliegende Erfindung kippte. Wenn nein, geht die Steuerung zum Schritt 3313 weiter. Anderenfalls wird überprüft 3311, ob die zugehörige Anzeige noch immer den unteren Rand des Fensters enthält. Wenn ja, geht die Steuerung zum Schritt 3313 weiter. Anderenfalls wird ein Schritt 3312 ausgeführt, der das Fenster nach links bewegt, so daß der rechte Rand auf dem zugeordneten Gerät bleibt. Dann wird überprüft 3313, ob die zugehörige Anzeige den linken Rand des Fensters enthielt, bevor der Benutzer die vorliegende Erfindung gekippt hat. Wenn nein, geht die Steuerung zum Schritt 3319 weiter. Anderenfalls wird überprüft 3314, ob der linke Rand des Fensters rechts von dem rechten Rand der zugehörigen Anzeige liegt. Wenn ja, geht die Steuerung zum Schritt 3316 weiter. Anderenfalls wird überprüft 3315, ob die zugehörige Anzeige den rechten Rand des Fensters enthielt, bevor der Benutzer die vorliegende Erfindung gekippt hat. Wenn nein, wird ein Schritt 3316 ausgeführt, der das Fenster nach links bewegt, so daß der linke Rand des Fensters einen konstanten Abstand zu dem zugehörigen rechten Rand der Anzeige einhält. Anderenfalls wird überprüft 3317, ob die zugehörige Grafikeinrichtung noch immer den linken Rand des Fensters enthält. Wenn ja, geht die Steuerung zum Schritt 3319 weiter. Anderenfalls wird ein Schritt 3318 ausgeführt, der das Fenster nach rechts bewegt, so daß der linke Rand auf der zugehörigen Anzeige bleibt. Dann wird überprüft 3319, ob die zugehörige Anzeige den oberen Rand des Fensters enthielt, bevor der Benutzer die Anzeige gekippt hat. Wenn nein, geht die Steuerung zum Schritt 3326 weiter. Anderenfalls wird überprüft 3320, ob der obere Rand des Fensters unter der unteren Kante der zugehörigen

Anzeige liegt. Wenn ja, geht die Steuerung zum Schritt 3322 weiter. Anderenfalls wird überprüft 3321, ob die zugehörige Anzeige den unteren Rand des Fensters enthielt, bevor der Benutzer die Anzeige gekippt hat. Wenn nein, wird ein Schritt 3322 ausgeführt, der das Fenster nach oben bewegt, so daß der obere Rand des Fensters einen konstanten Abstand zu der unteren Kante der zugehörigen Anzeige einhält. Anderenfalls wird überprüft 3323, ob die zugehörige Grafikeinrichtung noch immer den oberen Rand des Fensters enthält. Wenn ja, geht die Steuerung zum Schritt 3325 weiter. Anderenfalls wird ein Schritt 3325 ausgeführt, der das Fenster nach unten bewegt, so daß der obere Rand auf der zugehörigen Anzeige bleibt. Dann wird ein Schritt 3325 ausgeführt, der die vertikale Position des Fensters einstellt, so daß es unter dem Menübalken und, falls notwendig, einem Phantomfenster bleibt. Diese Routine ist verantwortlich für die Neupositionierung eines Fensters, nachdem der Benutzer eine Anzeige gekippt hat, so daß dieses zugänglich bleibt und bei der neuen Orientierung der Anzeigen weiterhin logisch plaziert ist.

Die in den Fig. 34a und 34b gezeigte Routine CheckResizeList 3401 überprüft 3402, ob es irgendwelche anhängigen Ereignisse gibt, welche die Anwendung bearbeiten muß. Wenn ja, geht die Steuerung zu dem aufrufenden Programm 3406 zurück. Anderenfalls wird überprüft 3403, ob "Maus nach unten"-Ereignisse aktiviert sind. Wenn nein, geht die Steuerung zum aufrufenden Programm 3406 zurück. Anderenfalls wird ein Schritt 3404 ausgeführt, der bei Bedarf eine neue Liste aus Fenstern aufbaut (in der umgekehrten Reihenfolge ihres Erscheinens auf dem Schreibtisch) und die Menügrößen der Anwendung neu berechnet. Dann wird eine Überprüfung 3405 durchgeführt, um sicherzustellen, daß die umgekehrte Fensterliste, welche momentan verarbeitet wird, für die Anwendung gilt, welche momentan ausgeführt wird. Wenn nein, geht die Steuerung zu dem aufrufenden Programm 3406 zurück. Anderenfalls wird überprüft 3407, ob es irgendwelche anhängigen

Ereignisse gibt, die die Anwendung bearbeiten muß. Wenn ja, geht die Steuerung zu dem aufrufenden Programm 3414 zurück. Anderenfalls wird überprüft 3404, ob das oberste Fenster, welches angezeigt wird, eine Dialogbox ist. Wenn ja, geht die Steuerung zum aufrufenden Programm 3414 zurück. Anderenfalls wird überprüft 3409, ob die umgekehrte Fensterliste leer ist. Wenn ja, wird ein Schritt 3412 ausgeführt, der den Finder anweist, falls notwendig, eine Aufräum-Operation durchzuführen. Anderenfalls wird ein Schritt 3410 ausgeführt, der das erste Fenster aus der umgekehrten Fensterliste entfernt. Dann wird überprüft 3411, ob das Fenster in dem System noch aktiv ist. Wenn nein, geht die Steuerung zu dem Schritt 3407 weiter. Anderenfalls wird ein Schritt 3413 ausgeführt, welcher die Größe des Fensters neu einstellt, falls dies notwendig ist. Schließlich geht die Steuerung zu dem aufrufenden Programm 3414 zurück. Diese Routine baut die Liste der Fenster, deren Größe neu eingestellt werden muß, nachdem der Benutzer eine Anzeige gemäß der vorliegenden Erfindung gekippt hat, auf und verarbeitet diese.

Die in Fig. 35 gezeigte Routine ResizeWindow 3501 führt zuerst einen Schritt 3502 aus, der die Variable portRect des Standes der Technik für das Fenster in globalen Koordinaten berechnet. Dann überprüft sie 3503, ob das Fenster in seinem Standardzustand ("Zoomed Out") war, bevor die Anzeige von dem Benutzer gekippt wurde. Wenn nein, geht die Steuerung zum Schritt 3505 weiter. Anderenfalls wird überprüft 3504, ob das Fenster noch immer in seinem Standardzustand ("Zoomed Out") ist. Wenn ja, geht die Steuerung zum Schritt 3505 weiter. Anderenfalls wird ein Schritt 3508 ausgeführt, der das Fenster vor alle anderen angezeigten Fenster zieht. Dann wird ein Schritt 3509 ausgeführt, der virtuelle Maustastenbetätigungen erzeugt, um die Fensteranwendung dazu zu bringen, dieses in seinen Standardzustand zurückzuzoomen. Dann geht die Steuerung zu dem aufrufenden Programm 3510 zurück. Ein Schritt 3505 berechnet den

Umfang, in dem die Größe des Fensters verändert werden muß, damit es auf der gekippten Anzeige vollständig zugänglich bleibt. Dann wird überprüft 3506, ob der Umfang der Größenveränderung nicht null ist. Wenn nein, geht die Steuerung zu dem aufrufenden Programm 3510 zurück. Anderenfalls wird ein Schritt 3507 ausgeführt, der virtuelle Maustastenbetätigungen erzeugt, welche zum Durchführen der Größenveränderungen benötigt werden. Schließlich geht die Steuerung zu dem aufrufenden Programm 3510 zurück. Diese Routine erzeugt die benötigten virtuellen Maustastenbetätigungen, falls es solche gibt, um die Größe eines Fensters neu einzustellen, nachdem der Benutzer eine Anzeige gemäß der vorliegenden Erfindung gekippt hat.

Die in Fig. 36 gezeigte Routine CalcWindResize 3601 führt zuerst einen Schritt 3602 aus, der das eingrenzende Rechteck der Anzeige zurückgibt, welche zu dem Fenster gehört, dessen Größe neu eingestellt wird. Dann wird ein Schritt 3603 ausgeführt, der lokale x- und y-Delta-Variablen nullt, welche dazu verwendet werden, den Umfang zu ermitteln, in dem die Größe des Fensters geändert werden muß. Dann wird überprüft 3604, ob der untere Rand des Fensters auf der zugehörigen Anzeige war, bevor der Benutzer die vorliegende Erfindung gekippt hat. Wenn nein, geht die Steuerung zum Schritt 3609 weiter. Anderenfalls wird überprüft 3605, ob die zugehörige Anzeige die Anzeige mit dem Menübalken ist. Wenn nein, geht die Steuerung zu dem Schritt 3607 weiter. Anderenfalls wird die minimale untere Versetzung für die Anwendung des Fensters von dem Rechteck der zugehörigen Anzeige in einem Schritt 3606 subtrahiert (die Versetzung, oder der Abstand, wird von dem Benutzer anwendungsweise bestimmt; die Voreinstellung ist null). Dann wird überprüft 3607, ob der untere Rand des Fensters noch immer das Rechteck der zugehörigen Anzeige schneidet. Wenn ja, geht die Steuerung zum Schritt 3609 weiter. Anderenfalls wird ein Schritt 3608 ausgeführt, der das Delta-y berechnet, welches notwendig ist, um den unteren Rand des Fensters auf der zugehörigen Anzeige zu halten. Dann

wird überprüft 3609, ob der untere Rand des Fensters auf der zugehörigen Anzeige war, bevor der Benutzer die Anzeige gekippt hat. Wenn nein, geht die Steuerung zum Schritt 3612 weiter. Dann wird überprüft 3610, ob der untere Rand des Fensters noch immer das Rechteck der zugehörigen Anzeige schneidet. Wenn ja, geht die Steuerung zum Schritt 3612 weiter. Anderenfalls wird die Prozedur 3611 aufgerufen, welche das Delta-x berechnet, welches benötigt wird, um den rechten Rand des Fensters auf der zugehörigen Anzeige zu halten. Schließlich wird der Schritt 3612 aufgerufen, welcher die globalen Koordinaten aus den lokalen Variablen Delta-x und Delta-y berechnet. Diese Routine ist verantwortlich für die Berechnung des Umfangs, in dem die Größe eines Fensters neu eingestellt werden muß, nachdem der Benutzer die Anzeige gekippt hat, so daß das Fenster bei der neuen Orientierung der Anzeige zugänglich und logisch plaziert bleibt.

Die in den Fig. 37a, 37b, 37c und 37d gezeigte Routine Rebuild Desktop 3701 führt zunächst einen Schritt 3702 aus, welcher das eingrenzende Rechteck der ersten Anzeige in dem Computersystem zurückgibt. Dann überprüft sie 3703, ob die Anzeige ihren oberen Rand mit einer anderen Anzeige teilte, bevor der Benutzer die vorliegende Erfindung gekippt hat. Wenn nein, geht die Steuerung zu der Überprüfung 3706 weiter. Anderenfalls wird geprüft 3704, ob die Anzeigen durch das Kippen getrennt wurden. Wenn nein, geht die Steuerung zu einem Schritt 3715 weiter. Anderenfalls wird ein Schritt 3705 ausgeführt, welcher den Nachbar nach unten bewegt, so daß die beiden Anzeigen wiederum denselben Rand haben, und die Steuerung geht zum Schritt 3715 weiter. Es wird geprüft 3706, ob die Anzeige einen gemeinsamen linken Rand mit einer anderen Anzeige hatte, bevor der Benutzer die Orientierung gemäß der vorliegenden Erfindung gekippt hat. Wenn nein, geht die Steuerung zu dem Schritt 3709 weiter. Anderenfalls wird überprüft 3707, ob die Anzeigen durch das Kippen getrennt wurden. Wenn nein, geht die Steuerung zum

Schritt 3715 weiter. Anderenfalls wird ein Schritt 3708 ausgeführt, der den Nachbarn nach rechts bewegt, so daß die beiden Anzeigen wiederum einen selben Rand teilen. Dann geht die Steuerung zum Schritt 3715 weiter. Es wird überprüft 3709, ob die Anzeige einen gemeinsamen unteren Rand mit einer anderen Anzeige hatte, bevor der Benutzer die vorliegende Erfindung gekippt hat. Wenn nein, geht die Steuerung zum Schritt 3712 weiter. Anderenfalls wird überprüft 3710, ob die Anzeigen durch das Kippen getrennt wurden. Wenn nein, geht die Steuerung zum Schritt 3715 weiter. Anderenfalls wird ein Schritt 3711 aufgerufen, welcher den Nachbarn nach oben bewegt, so daß die beiden Anzeigen wiederum einen gemeinsamen Rand haben, und die Steuerung geht zum Schritt 3715 weiter. Wenn die Überprüfung 3709 ergibt, daß es am unteren Rand keinen Nachbarn gibt, wird überprüft 3712, ob die Anzeige den gleichen rechten Rand wie eine andere Anzeige hatte, bevor der Benutzer die vorliegende Erfindung gekippt hat. Wenn nein, geht die Steuerung zum Schritt 3715 weiter. Anderenfalls wird überprüft 3713, ob die Anzeigen durch das Kippen getrennt wurden. Wenn nein, geht die Steuerung zum Schritt 3715 weiter. Anderenfalls wird ein Schritt 3714 aufgerufen, der den Nachbarn nach links bewegt, so daß die beiden Anzeigen wiederum denselben Rand gemeinsam haben. Danach wird überprüft 3715, ob irgendwelche Anzeigen bewegt wurden, um ihre Ränder neu zu verbinden. Wenn ja, geht die Steuerung zum Schritt 3702 weiter. Anderenfalls wird eine Prozedur 3716 aufgerufen, welche die nächste Anzeige in dem System zurückgibt. Dann wird eine Überprüfung 3717 durchgeführt, um sicherzustellen, daß von dem Schritt 3716 eine Anzeige zurückgegeben wurde. Wenn ja, geht die Steuerung zum Schritt 3703 weiter. Anderenfalls wird ein Schritt 3718 ausgeführt, der die lokale Variable CurrentGD als die erste Anzeige in dem System festlegt. Dann wird ein Schritt 3719 ausgeführt, um die lokale Variable CompareGD als die erste Anzeige in dem System einzustellen. Dann wird überprüft 3720, ob die CurrentGD-Anzeige und die CompareGD-Anzeige übereinstimmen.

Wenn ja, geht die Steuerung zum Schritt 3725 weiter. Anderenfalls wird überprüft 3721, ob die beiden Anzeigen sich im Koordinatenraum überdecken. Wenn nein, geht die Steuerung zum Schritt 3725 weiter. Anderenfalls wird überprüft 3722, ob es einfacher ist, die CompareGD-Anzeige nach rechts zu bewegen. Wenn ja, wird ein Schritt 3723 ausgeführt, um die CompareGD-Anzeige nach rechts zu bewegen, so daß die beiden Anzeigen den Koordinatenraum nur längs eines Randes teilen, vorausgesetzt, daß die CompareGD-Anzeige nicht der linke Nachbar der CurrentGD-Anzeige ist und daß die CurrentGD-Anzeige nicht der rechte Nachbar der CompareGD-Anzeige ist. Anderenfalls wird ein Schritt 3724 ausgeführt, um die CompareGD-Anzeige nach unten zu bewegen, so daß die beiden Anzeigen den Koordinatenraum nur längs einer Kante teilen, vorausgesetzt, daß die CompareGD-Anzeige nicht der obere Nachbar der CurrentGD-Anzeige ist und daß die CurrentGD-Anzeige nicht der untere Nachbar der CompareGD-Anzeige ist. In jeden Fall wird ein Schritt 3725 aufgerufen, welcher der lokalen Variablen CompareGD die nächste Anzeige in dem System nach ihrem momentanen Wert zuordnet. Dann wird eine Überprüfung 3726 durchgeführt, um sicherzustellen, daß CompareGD gültig ist (daß es tatsächlich eine weitere Anzeige in dem System gibt). Wenn ja, geht die Steuerung zum Schritt 3720 weiter. Wenn nein, wird überprüft 3727, ob irgendwelche Anzeigen bewegt wurden (ihre Koordinatensysteme änderten). Wenn ja, geht die Steuerung zum Schritt 3718 weiter. Wenn nein, wird ein Schritt 3728 ausgeführt, welcher der lokalen Variablen CurrentGD die nächste Anzeige in dem System nach ihrem momentanen Wert zuordnet. Dann wird eine Überprüfung 3729 durchgeführt, um sicherzustellen, daß CurrentGD gültig ist (daß es tatsächlich eine weitere Anzeige in dem System gibt). Wenn ja, geht die Steuerung zum Schritt 3719 weiter. Anderenfalls wird ein Schritt 3730 ausgeführt, der die Koordinaten aller Systeme in der Anzeige normiert, so daß die obere linke Ecke der Anzeige mit dem Menübalken 0,0 ist. Schließlich geht die Ausführung zu dem aufrufenden Programm 3731 zurück. Diese Prozedur ist

verantwortlich für die Neuanordnung der Koordinatensysteme der Anzeigen, nachdem der Benutzer eine Anzeige gemäß der vorliegenden Erfindung gekippt hat.

Die in Fig. 38 gezeigte Routine CleanUpFinder 3801 überprüft 3802, ob dies der erste Durchgang durch die Routine ist, seitdem der Benutzer eine Anzeige gemäß der vorliegenden Erfindung gekippt hat. Wenn nein, geht die Steuerung zum Schritt 3803 weiter. Wenn ja, wird überprüft 3805, ob die gekippte Anzeige den Menübalken enthielt. Wenn nein, geht die Steuerung zum Schritt 3809 weiter. Anderenfalls wird ein Schritt 3806 ausgeführt, welcher alle sichtbaren Fenster des Finders verbirgt. Dann geht die Ausführung zu dem aufrufenden Programm 3810 zurück. Es wird überprüft 3803, ob dies der zweite Durchgang durch die Routine ist, seitdem der Benutzer eine Anzeige gemäß der vorliegenden Erfindung gekippt hat. Wenn nein, geht die Steuerung zum Schritt 3804 weiter. Wenn ja, wird ein Schritt 3807 ausgeführt, der die virtuellen Maustastenbetätigungen erzeugt, die benötigt werden, um den Finder dazu zu bringen, seine Aufräumoperation zu beginnen. Dann geht die Ausführung zu dem aufrufenden Programm 3810 zurück. Es wird überprüft 3804, ob dies der dritte Durchgang durch die Routine ist, seitdem der Benutzer eine Anzeige gemäß der vorliegenden Erfindung gekippt hat. Wenn nein, geht die Steuerung zum Schritt 3809 weiter. Anderenfalls wird ein Schritt 3808 ausgeführt, der alle Fenster des Finders zeigt, welche zuvor von dem Schritt 3806 verborgen wurden. Dann wird ein Schritt 3809 ausgeführt, welcher die vorliegende Erfindung dazu bringt, ihren Finder-Aufräummodus zu verlassen. Schließlich kehrt die Ausführung zu dem aufrufenden Programm 3810 zurück. Diese Prozedur koordiniert die Aktivitäten, welche dazu benötigt werden, den Finder dazu zu bringen seine Schreibtischpiktogramme neu zu positionieren, nachdem der Benutzer eine Anzeige gekippt hat. Sie wird so lange aufgerufen, wie die vorliegende Erfindung in ihrem Finder-Aufräummodus ist.

Die in Fig. 39 gezeigte Routine MapBigRect 3901 ruft zuerst einen Schritt 3902 auf, um die ersten Daten (Aufzeichnungen) in einer Liste mit möglichen Größen zu erhalten, welche die Anzeige mit dem Menübalken haben kann. Dann wird ein Schritt 3903 ausgeführt, welcher ein temporäres Rechteck aus den Grenzen gemäß der Anzeigegrößen-Daten bildet. Dann vergrößert ein Schritt 3904 das temporäre Rechteck um zwei Pixel auf jeder Seite. Danach wird überprüft 3905, ob das von der aufrufenden Routine übergebene Rechteck in das vergrößerte temporäre Rechteck paßt. Wenn nein, geht die Steuerung zum Schritt 3910 weiter. Andererfalls stellt ein Schritt 3906 den oberen Rand des temporären Rechtecks als die Höhe des Menübalkens ein. Dann wird ein Schritt 3907 ausgeführt, um das Rechteck um 8 Pixel auf der linken Seite, 8 Pixel auf der rechten Seite, 64 oben und 64 unten zu verkleinern. Dann wird ein Schritt 3908 ausgeführt, um den rechten Rand des temporären Rechtecks um 80 Pixel zu bewegen. Dann wird überprüft 3909, ob das temporäre Rechteck in das übergebene Rechteck paßt. Wenn ja, modifiziert ein Schritt 3912 die untere rechte Ecke des übergebenen Rechtecks, so daß die untere rechte Ecke ihren Abstand im Verhältnis zur momentanen unteren rechten Ecke der Anzeige mit dem Menübalken beibehält. Dann geht die Steuerung zu dem aufrufenden Programm 3913 zurück. Der Schritt 3910 holt die nächsten Daten in der Liste aus möglichen Größen, welche die Anzeige mit dem Menübalken haben kann. Dann wird eine Überprüfung 3911 ausgeführt, um sicherzustellen, daß die gerade aufgerufenen Daten existieren. Wenn ja, geht die Ausführung zum Schritt 3903 weiter. Anderenfalls kehrt die Steuerung zu dem aufrufenden Programm 3913 zurück. Diese Prozedur bildet ein gegebenes Rechteck ab (welches üblicherweise den Standardzustand eines Fensters wiedergibt), damit es möglichst gut in die aktuellen Dimensionen der Anzeige mit dem Menübalken paßt.

Die in Fig. 40 gezeigte Routine ProcessNewWind 4001 führt zuerst einen Schritt 4002 aus, welcher eine Hilfsdatenaufzeich-

nung für das Fenster erzeugt, so daß die vorliegende Erfindung bestimmte zusätzliche Daten verfolgen kann, welche das Fenster betreffen. Dann wird überprüft 4003, ob das Fenster vergrößerbar ist. Wenn nein, geht die Steuerung zu dem aufrufenden Programm 4009 zurück. Wenn ja, wird überprüft 4004, ob das Fenster in die momentanen Grenzen der Anzeige mit dem Menübalken paßt. Wenn ja, geht die Steuerung zu dem aufrufenden Programm 4009 zurück. Wenn nein, wird überprüft 4005, ob das Fenster logisch auf die momentanen Abmessungen der Anzeige mit dem Menübalken abgebildet werden kann. Wenn nein, geht die Steuerung zu dem aufrufenden Programm 4009 zurück. Wenn ja, wird überprüft 4006, ob das Fenster sichtbar ist. Wenn ja, wird ein Schritt 4007 ausgeführt, welcher virtuelle Maustastenbetätigungen erzeugt, um die Größe des Fensters neu einzustellen, so daß es in die momentanen Dimensionen der Anzeige mit dem Menübalken paßt. Wenn nein, wird ein Schritt 4008 ausgeführt, welcher das Fenster an die Liste mit Fenstern, deren Größe neu eingestellt werden soll, für die spätere Verwendung hinzufügt. In jedem Fall geht die Steuerung danach als nächstes zu dem aufrufenden Programm 4009 zurück. Diese Routine führt zwei Funktionen aus. Zunächst reserviert sie die zusätzliche Datenstruktur, welche verwendet wird, um Schlüsselinformationen zu verfolgen, welche bei der Neuorientierung/Neueinstellung der Größe eines Fensters benötigt wird, nachdem der Benutzer eine Anzeige gekippt hat. Zweitens überprüft sie die Größe des Fensters, um sicherzustellen, daß seine Größe einer vorherigen Abmessung der Anzeige entspricht, zu welcher es gehört. Wenn es nicht die richtige Größe hat, werden Maustastenbetätigungen erzeugt, um die Größe des Fensters auf die momentanen Abmessungen der zugehörigen Anzeige neu einzustellen.

Die in den Fig. 41a und 41b gezeigte Routine GetWindGD 4101 führt zunächst einen Schritt 4102 durch, welcher ein Rechteck zurückgibt, daß die Koordinaten des sichtbaren Teils des Fensters, übersetzt in das Koordinatensystem der Anzeige enthält.

Dann wird ein Schritt 4103 ausgeführt, welcher die Koordinaten von der ersten Anzeigevorrichtung in dem System holt. Dann wird mit den Rechtecken, welche das Fenster und die Anzeige wiedergeben, eine Überprüfung 4104 ausgeführt, welche ermittelt, ob die Anzeigevorrichtung den oberen Rand des Fensters enthält. Wenn nein, geht die Steuerung zum Schritt 4106 weiter. Anderenfalls wird ein Schritt 4105 ausgeführt, der die Größe acht zu einer gewichteten Summe des Fensters addiert, um zu ermitteln, welche Anzeige dem Fenster zugeordnet wird. Dann wird überprüft 4106, ob die Anzeigevorrichtung den linken Rand des Fensters enthält. Wenn nein, geht die Steuerung zum Schritt 4108 weiter. Anderenfalls wird ein Schritt 4107 ausgeführt, der die Größe vier zu der gewichteten Summe addiert. Dann wird überprüft 4108, ob die Vorrichtung den unteren Rand des Fensters enthält. Wenn nein, geht die Steuerung zum Schritt 4110 weiter. Anderenfalls wird der Schritt 4109 ausgeführt, der die Größe zwei zu der gewichteten Summe addiert. Dann wird überprüft 4110, ob die Vorrichtung den rechten Rand des Fensters enthält. Wenn nein, geht die Steuerung zum Schritt 4112 weiter. Anderenfalls wird der Schritt 4111 aufgerufen, der die Größe eins zu der gewichteten Summe addiert. Dann wird überprüft 4112, ob die gewichtete Summe für diese Vorrichtung die größte bisher gefundene ist. Wenn nein, geht die Steuerung zum Schritt 4114 weiter. Anderenfalls speichert der Schritt 4113 das Ergebnis und die zugehörige Vorrichtung in der Hilfsdatenstruktur für das Fenster. Dann wird überprüft 4114, ob es noch weitere Anzeigevorrichtungen in dem System gibt. Wenn nein, geht die Steuerung zum Schritt 4116 weiter. Anderenfalls holt der Schritt 4115 die Grenzen der nächsten Anzeige in dem System. Die Steuerung geht dann zum Schritt 4104 weiter. Es wird überprüft 4116, ob das höchste erhaltene Ergebnis null war. Wenn nein, geht die Steuerung zum Schritt 4118 weiter. Anderenfalls wird der Schritt 4117 ausgeführt, welcher die Anzeige, welche den Menübalken enthält, und das Fenster zusammenbringt. Dann wird der Schritt 4118 ausgeführt, der die Fenster-Versetzungen von der oberen linken und

der unteren rechten Ecke der zugehörigen Anzeige berechnet. Schließlich geht die Steuerung zu dem aufrufenden Programm 4119 zurück. Diese Routine ermittelt, welche Anzeige am besten dem Fenster zugeordnet wird, wodurch das relative Erscheinungsbild des Fensters besser aufrechterhalten werden kann, wenn der Benutzer eine Anzeige gemäß der vorliegenden Erfindung kippt. Die Anzeige, welche die obere, linke, untere und rechte Ecke des Fensters enthält, wird in dieser Reihenfolge gewichtet.

Obwohl die bevorzugte Ausführungsform mit definierten Fenstern auf einem Computeranzeigesystem mit zwei Bildschirmen arbeitet, ist die Erfindung auch dann nützlich, wenn ein Bild auf einer oder mehreren Anzeigen beliebiger Art, deren Orientierung nicht völlig unveränderlich ist, dargestellt werden kann.

Es wird somit ein Verfahren vorgesehen, mit dem Fenster einer Computeranzeige bewegt und ihre Größe neu eingestellt werden kann, wenn ein Computerbildschirm zwischen einer Porträt-Orientierung und einer Landschafts-Orientierung gekippt wird.

691 23 465.5-08
Radius Inc.
RL 2358

1. Verfahren zum Vorsehen einer Anzeige für einen Computer, der eine Anzeigevorrichtung (34, 41) verwendet, die wenigstens ein Anzeigefenster (10, 11) gemäß einer Gruppe von Regeln vorsieht, g e k e n n z e i c h n e t durch selektives Positionieren der Anzeigevorrichtung (34, 41) in einer von mehreren physischen Orientierungen (1, 2), Erfassen (3202) einer Veränderung der physischen Orientierung (1, 2) der Anzeigevorrichtung (34, 41), Ermitteln eines Koordinatensystems (42, 44), welches Positionen auf der Anzeigevorrichtung (41) in der veränderten Orientierung (2) beschreibt, Erzeugen (3207) einer Desktop-Einrichtung zum Erzeugen des Anzeigefensters gemäß diesem Koordinatensystem, Aktualisieren (3208, 3210) von Positions- und Größenmerkmalen des Anzeigefensters (10, 11) nach Maßgabe des Koordinatensystems, Aktualisieren (3211) von Grenzen für die Curser- und Mausbewegung nach Maßgabe des Koordinatensystems und erneutes Darstellen (3212) des Anzeigefensters auf der Anzeigevorrichtung (2).

2. Verfahren nach Anspruch 1, dadurch g e k e n n z e i c h n e t, daß die Regelgruppe eine Regel zum Beibehalten des relativen Abstandes von einer ausgewählten Position auf dem Fenster (10, 11) zu einer ausgewählten Position (24) auf der Anzeigevorrichtung aufweist.

3. Verfahren nach Anspruch 1 oder 2, dadurch g e k e n n z e i c h n e t, daß die Anzeige eine Computerbildanzeige umfaßt, welche auf mehreren Anzeigeschirmen (34, 35) liegt, und daß die Regelgruppe eine Regel zum Ermitteln, welcher der mehreren Anzeigeschirme (34, 35) das Fenster (38) kontrolliert, umfaßt.

4. Verfahren nach Anspruch 3, dadurch g e k e n n z e i c h n e t, daß das Bewegen und Verändern der Größe des Fensters (38) abhängig von der Änderung der Orientierung des Anzeigeschirms (35) ist, der das Fenster (38) kontrolliert.

5. Verfahren nach Anspruch 3 oder 4, dadurch g e k e n n z e i c h n e t, daß der Anzeigeschirm (35), der das Fenster (38) kontrolliert, ermittelt wird, indem die Wichtigkeit verschiedener Ränder des Fensters (38) gewichtet wird, eine gewichtete Summe für jeden Anzeigeschirm (35) berechnet wird, auf dem ein Teil des Fensters (38) erscheint, und die Kontrolle des Fensters (38) dem Anzeigeschirm (35) mit der größten gewichteten Summe zugewiesen wird.

6. Verfahren nach Anspruch 5, dadurch g e k e n n z e i c h n e t, daß die gewichtete Summe berechnet wird, indem 8 zu der gewichteten Summe eines Anzeigeschirms (35) addiert wird, wenn dieser Anzeigeschirm (35) den oberen Rand des Fensters (38) enthält, 4 zu der gewichteten Summe des Anzeigeschirms (35) addiert wird, wenn dieser Anzeigeschirm (35) den linken Rand des Fensters (38) enthält, 2 zu der gewichteten Summe eines Anzeigeschirms (35) addiert wird, wenn dieser Anzeigeschirm (35) den rechten Rand des Fensters (38) enthält, und 1 zu der gewichteten Summe eines Anzeigeschirms (35) addiert wird, wenn dieser Anzeigeschirm (35) den unteren Rand des Fensters (38) enthält.

7. Verfahren nach Anspruch 5 oder 6, dadurch g e k e n n z e i c h n e t, daß die Position eines Fensters (38), das von einem Anzeigeschirm (35) kontrolliert wird, bei einer Änderung der Orientierung eines der anderen

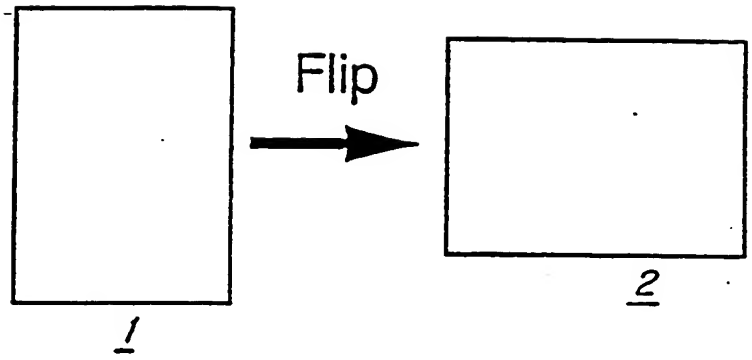
Anzeigeschirme (34, 35) relativ zu jenem Anzeigeschirm (35) unverändert bleibt.

8. Verfahren nach einem der vorangehenden Ansprüche, dadurch g e k e n n z e i c h n e t, daß die Regelgruppe eine Regel enthält, gemäß der eine erste Änderung der Orientierung des Anzeigeschirms (1) gefolgt von einer zweiten Änderung der Orientierung des Anzeigeschirms (1) zurück zu der Anfangsorientierung des Anzeigeschirms zur Folge hat, daß das Fenster (11) der Anzeige dieselbe Position und Größe wie vor der ersten Änderung der Orientierung hat.

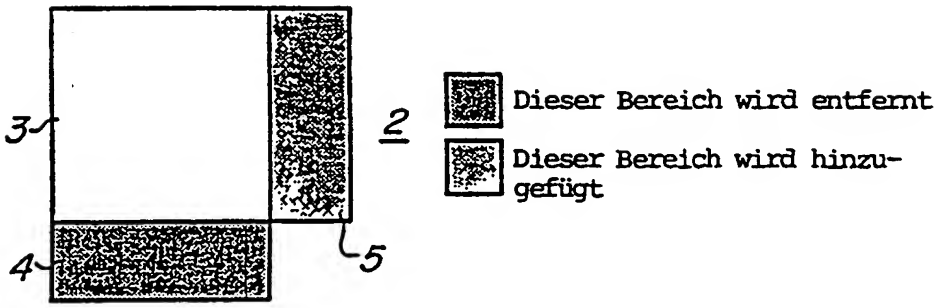
9. Verfahren nach einem der vorangehenden Ansprüche, dadurch g e k e n n z e i c h n e t, daß die Regelgruppe eine Regel enthält, bei der das Fenster (11) der Anzeige abhängig von einer Änderung der Orientierung des Anzeigeschirms (1) automatisch eine neue Größe enthält, um den neu ausgerichteten Anzeigschirm (2) mit dem Fenster (11) der Anzeige vollständiger auszufüllen.

10. Verfahren nach einem der vorangehenden Ansprüche, dadurch g e k e n n z e i c h n e t, daß das Fenster (11) der Anzeige bewegt und seine Größe verändert wird, indem Computercodeanweisungen ausgeführt werden, welche abhängig davon sind, daß ein Benutzer das Fenster (11) mittels eines Zeigers bewegt und seine Größe verändert.

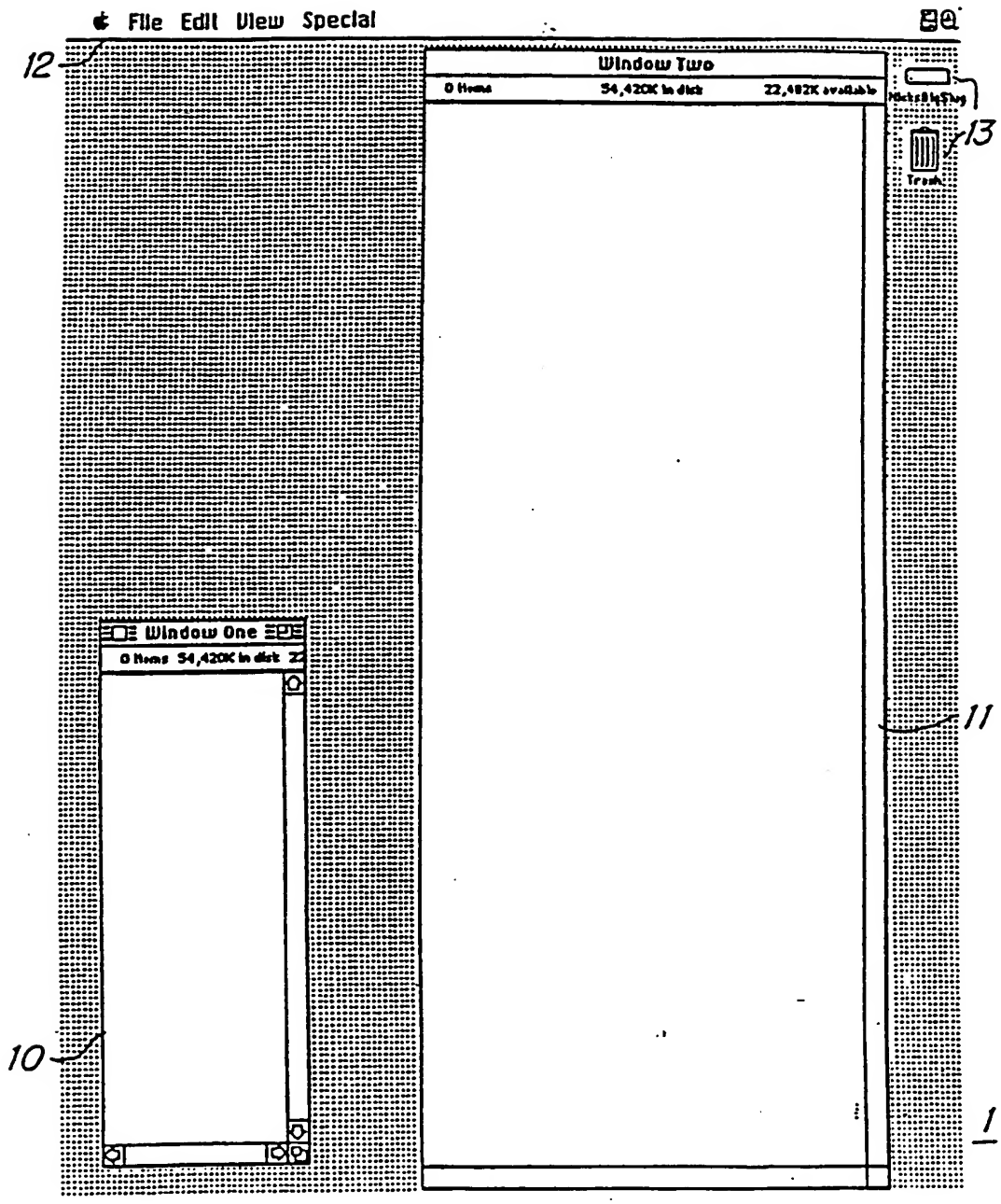
11. Verfahren nach einem der vorangehenden Ansprüche, dadurch g e k e n n z e i c h n e t, daß die Regelgruppe mehrere Strategien zum Bewegen und Verändern der Größe des Fensters der Anzeige umfassen, abhängig von der Identifikation der Art des Computerprogramms, welches das Fenster der Anzeige erzeugt.



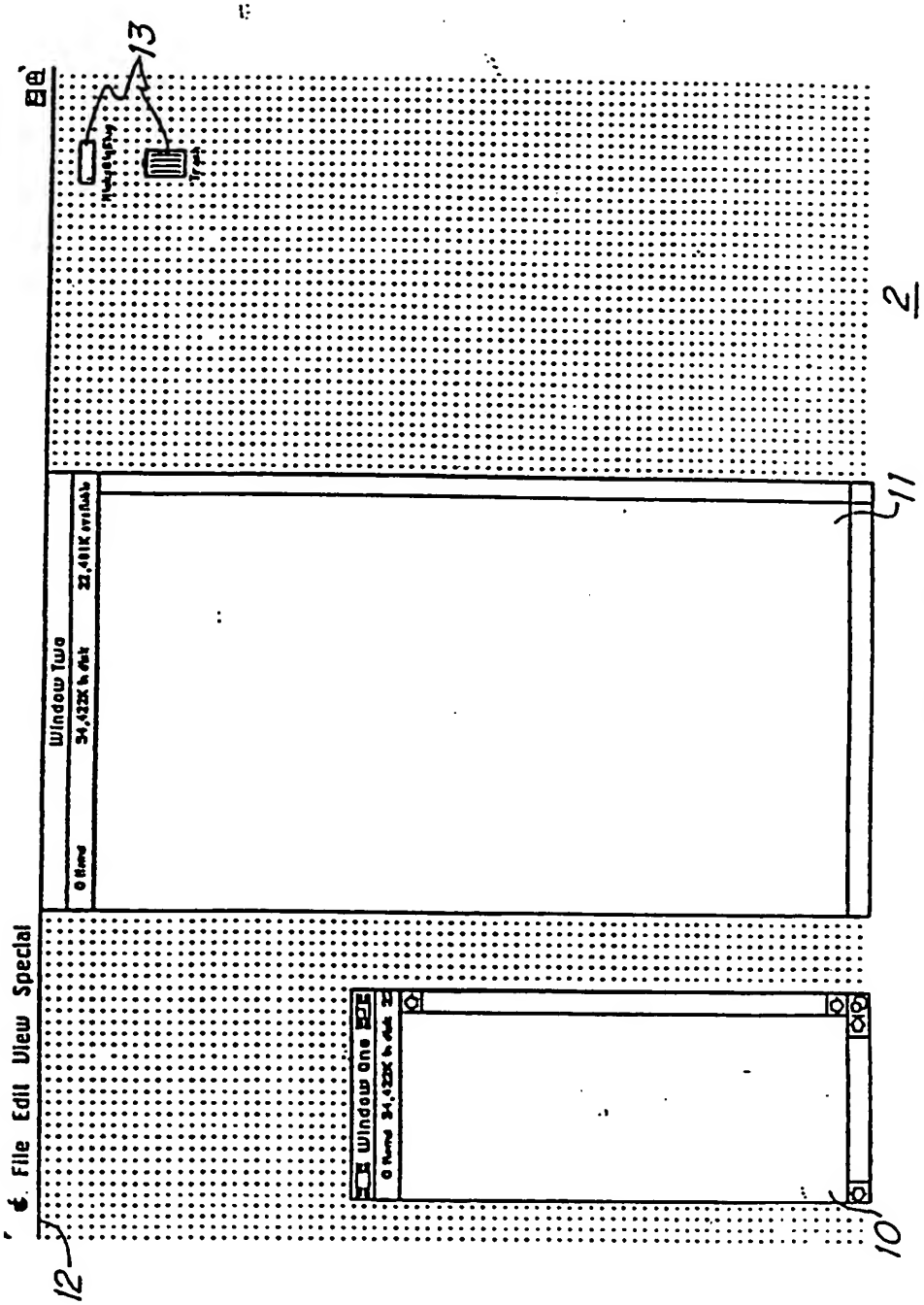
Figur 1a



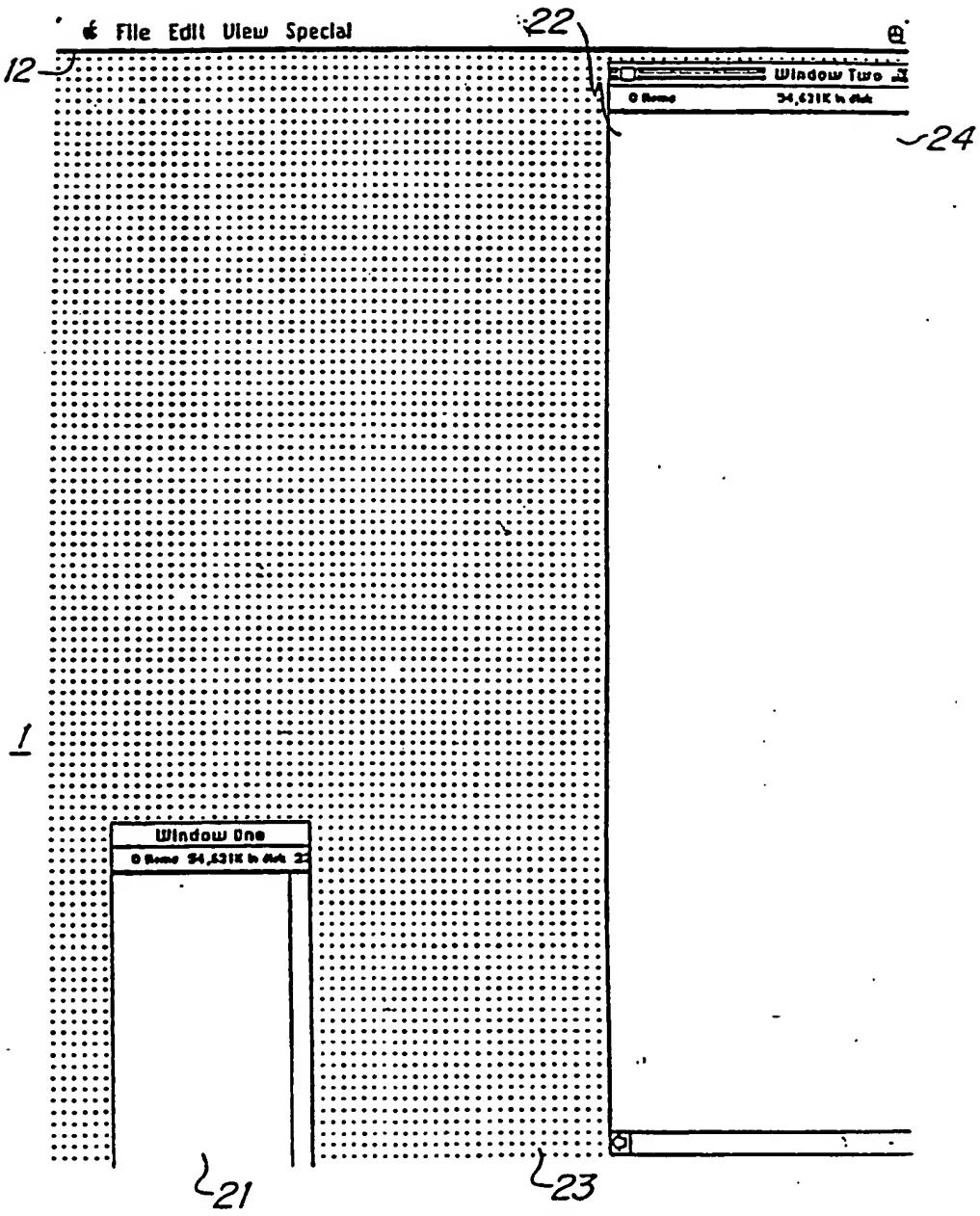
Figur 1b


File Edit View Special
12
Window Two
0 Items
13
Trash
Window One
0 Items 54,420K in disk
11
10
1

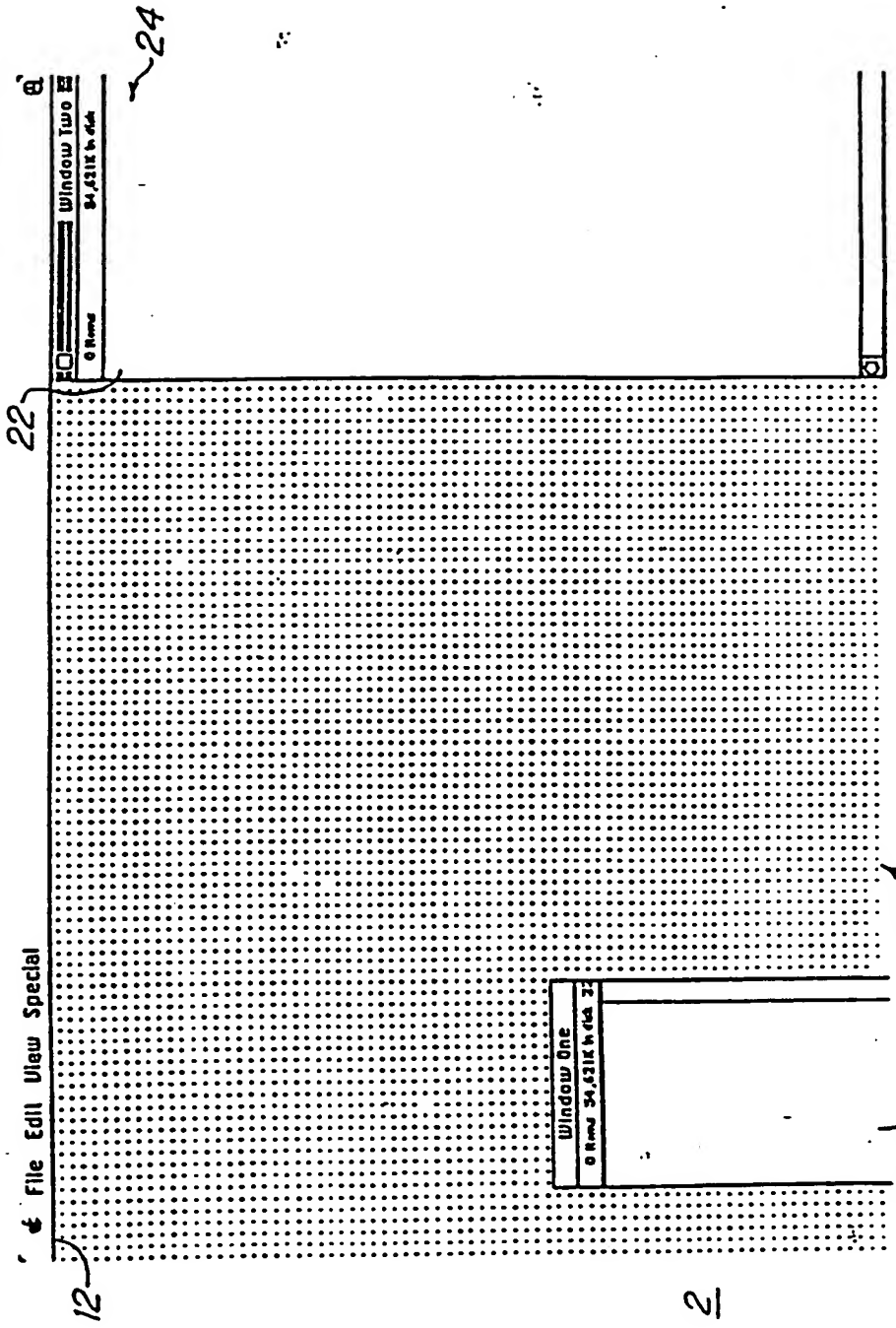
Figur 2



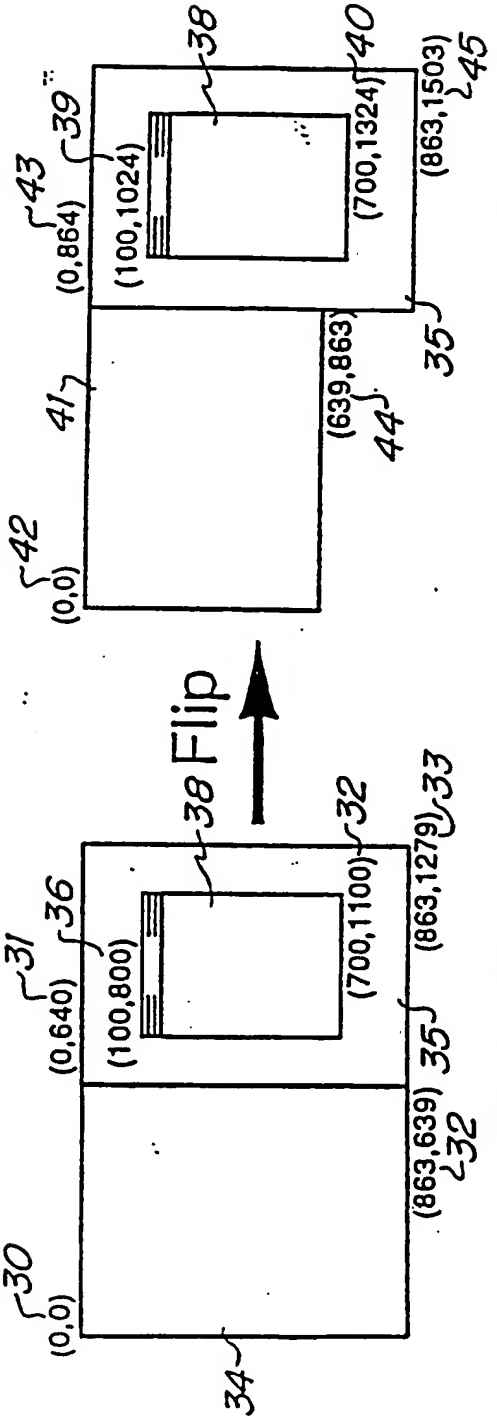
Figur 3



Figur 4

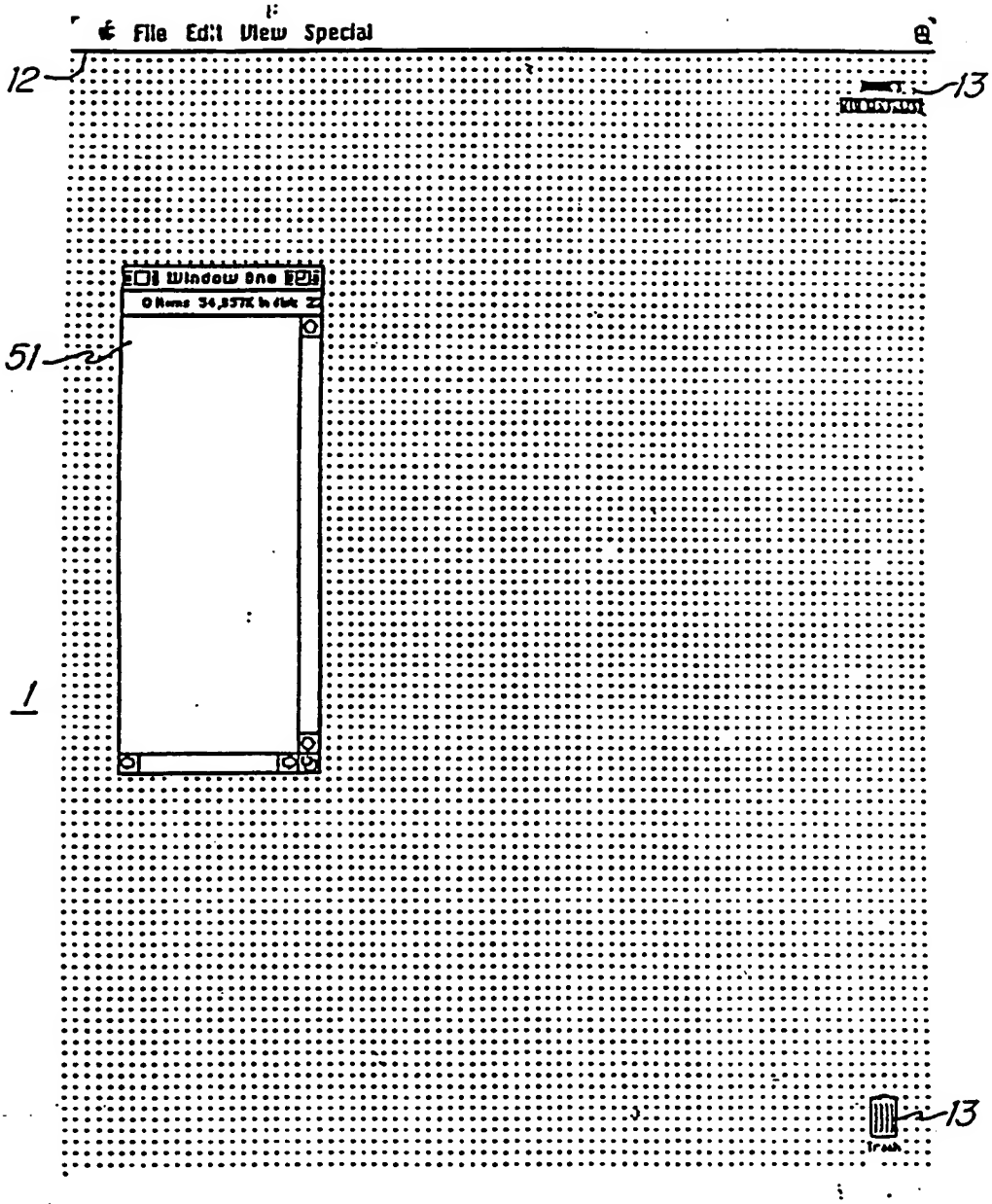


Figur 5

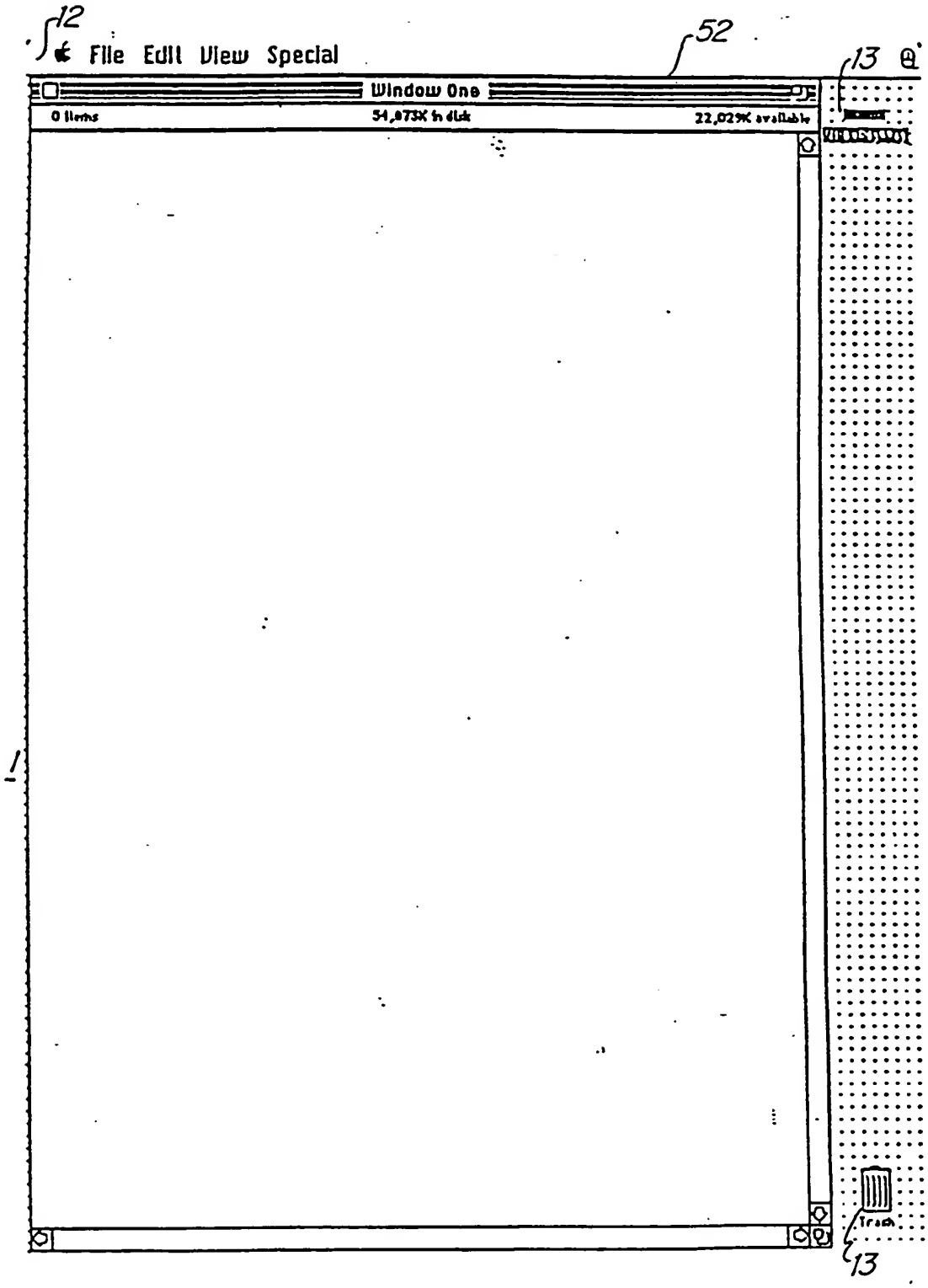


Figur 6a

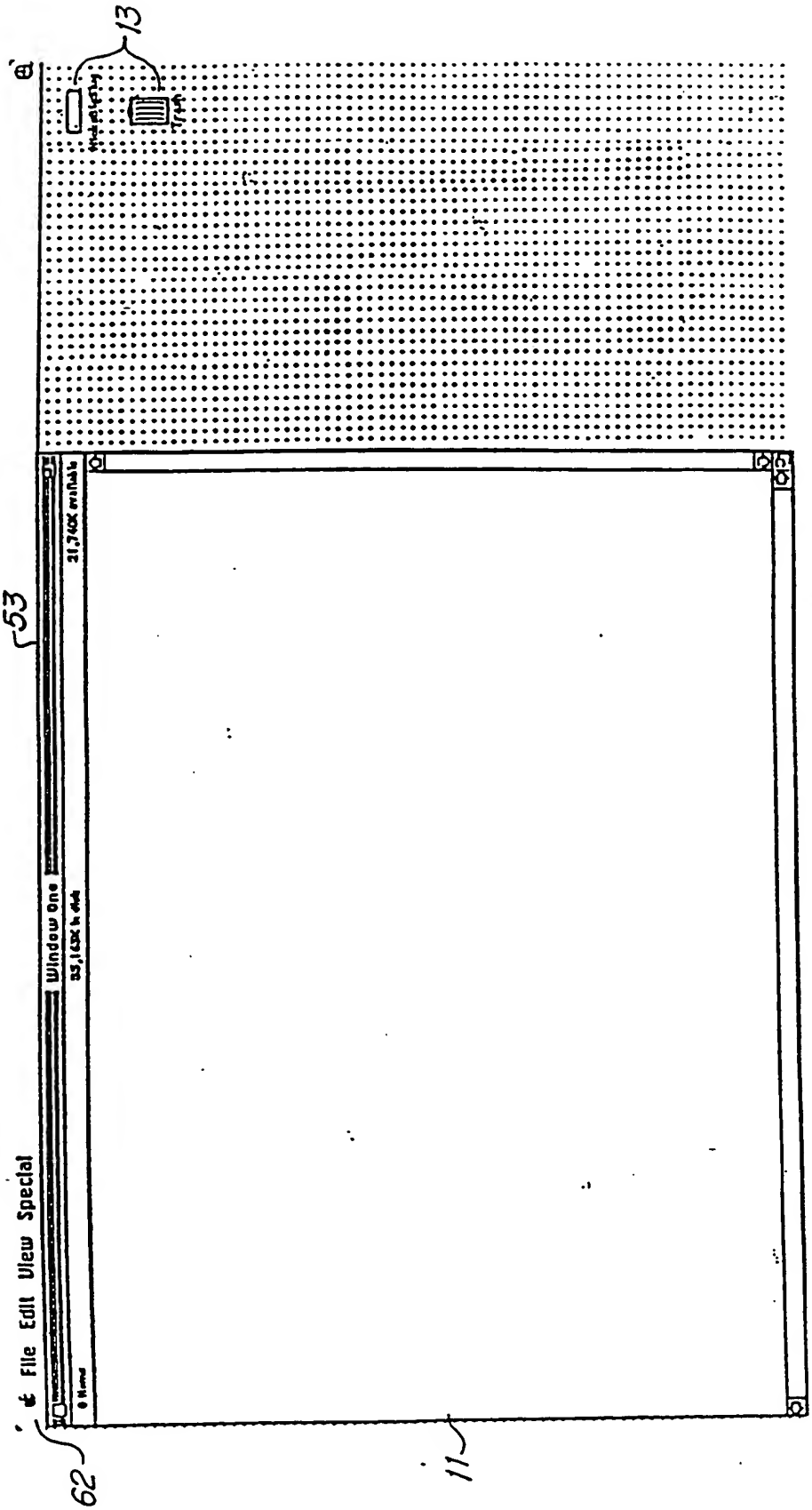
Figur 6b



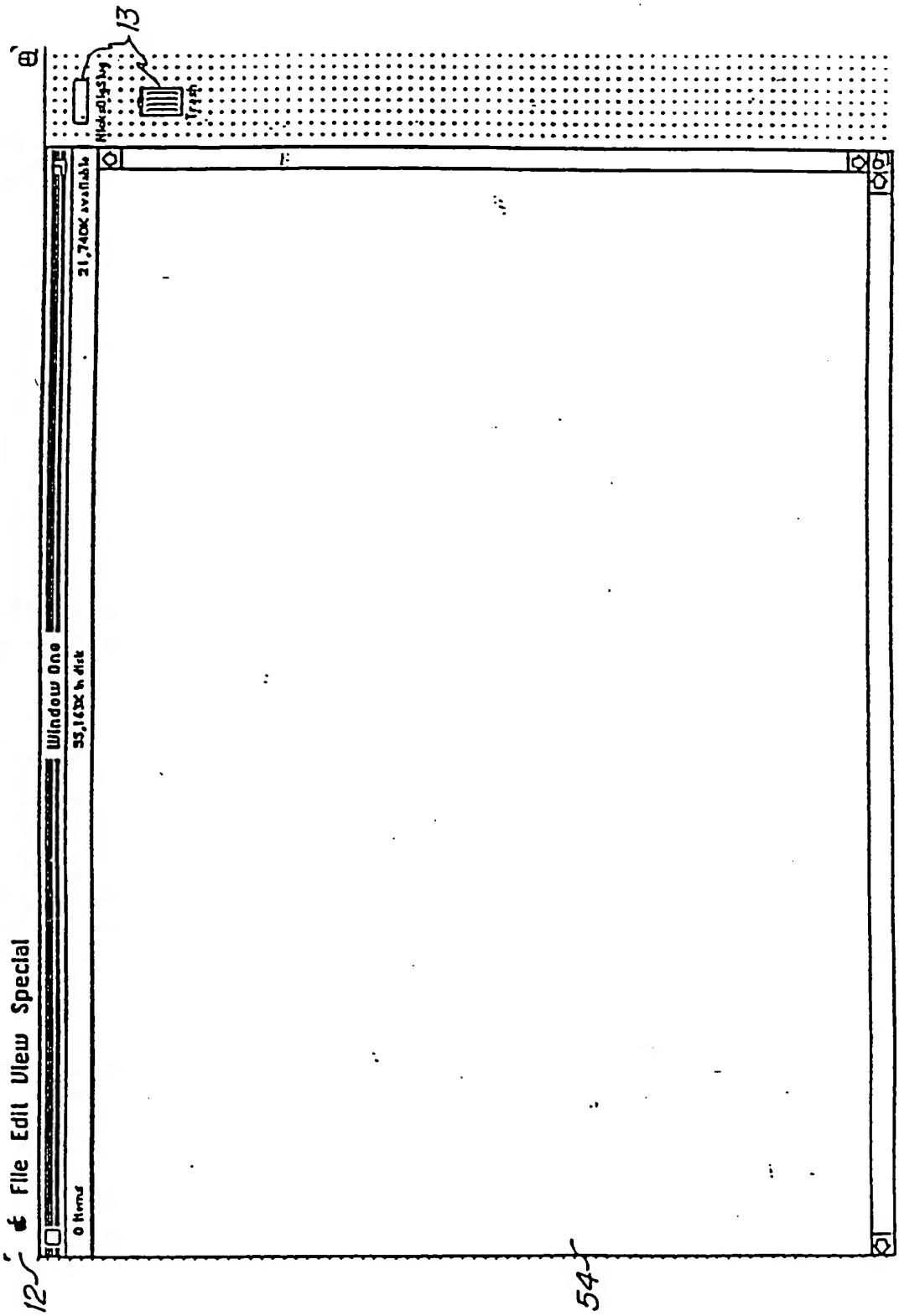
Figur 7


12
File Edit View Special
52
13
Window One
0 items
54,873K in disk
22,029K available
1
13

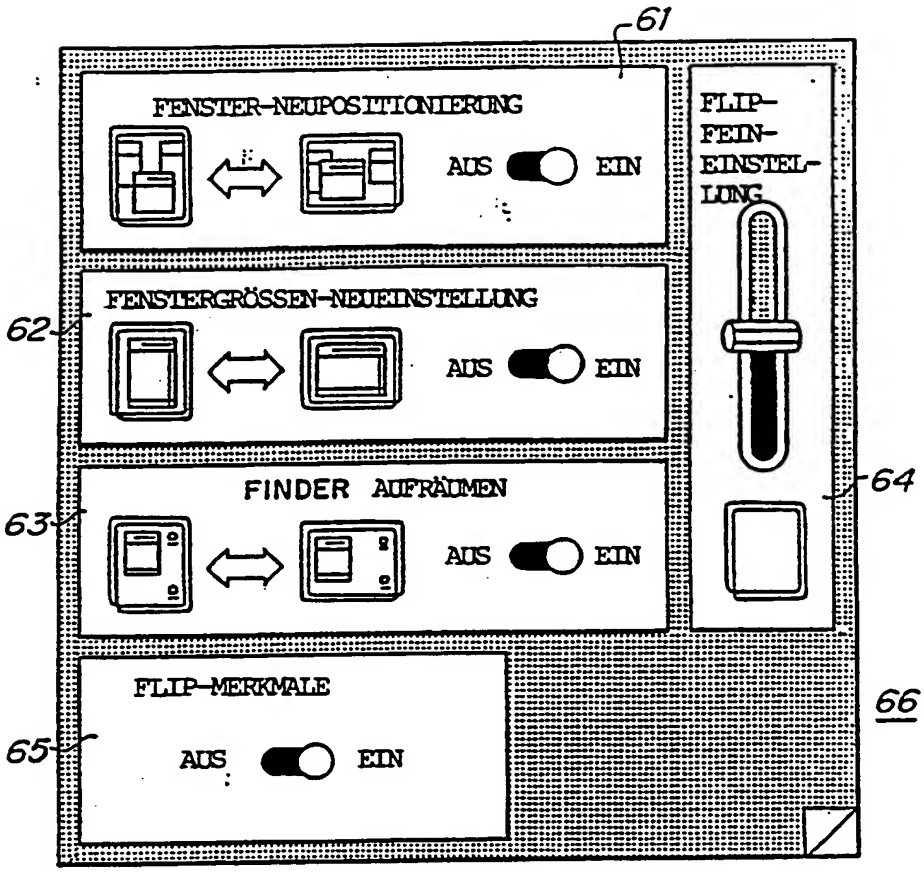
Figur 8



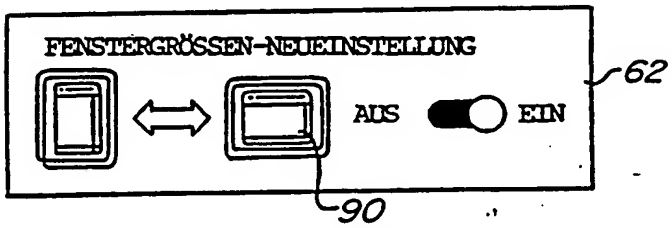
Figur 9


13
File Edit View Special
Window One
21,740K available
0 items
12
54

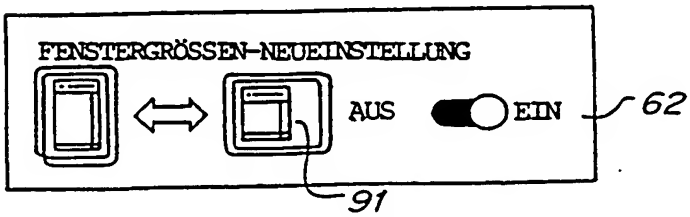
Figur 10



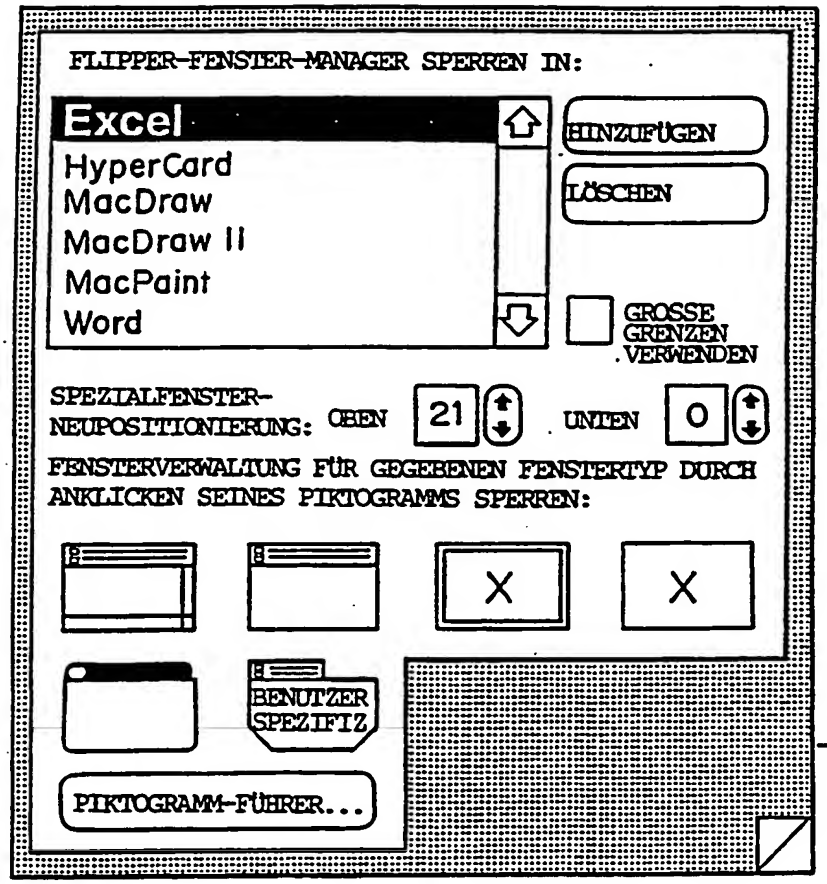
Figur 11



Figur 11a



Figur 11b


FLIPPER-FENSTER-MANAGER SPERREN IN:
Excel
HyperCard
MacDraw
MacDraw II
MacPaint
Word
HINZUFÜGEN
LÖSCHEN
GROSSE GRENZEN VERWENDEN
SPEZIALFENSTER-NEUPOSITIONIERUNG: OBEN 21 UNTEN 0
FENSTERVERWALTUNG FÜR GEGEBENEN FENSTERTYP DURCH ANKLICKEN SEINES PIKTOGRAMMS SPERREN:
X
X
BENUTZER SPEZIFIZ
PIKTOGRAMM-FÜHRER...
71

Figur 12

(STAND DER TECHNIK)

TRAPS MIT ROM KORRIGIERT

Slot Manager

101

UNTERSTÜTZUNGSROUTINEN

VERSCHIEDENES

119

(STAND DER TECHNIK)

TRAPS MIT INIT KORRIGIERT

Get Next Event

102

Init Graf

103

New Window

104

Init Windows

105

Select Window

106

Close Window

107

Drag Window

108

Show Hide

109

Grow Window

110

Size Window

111

Zoom Window

112

Menu Select

113

Track Box

114

Get Mouse

115

Button

116

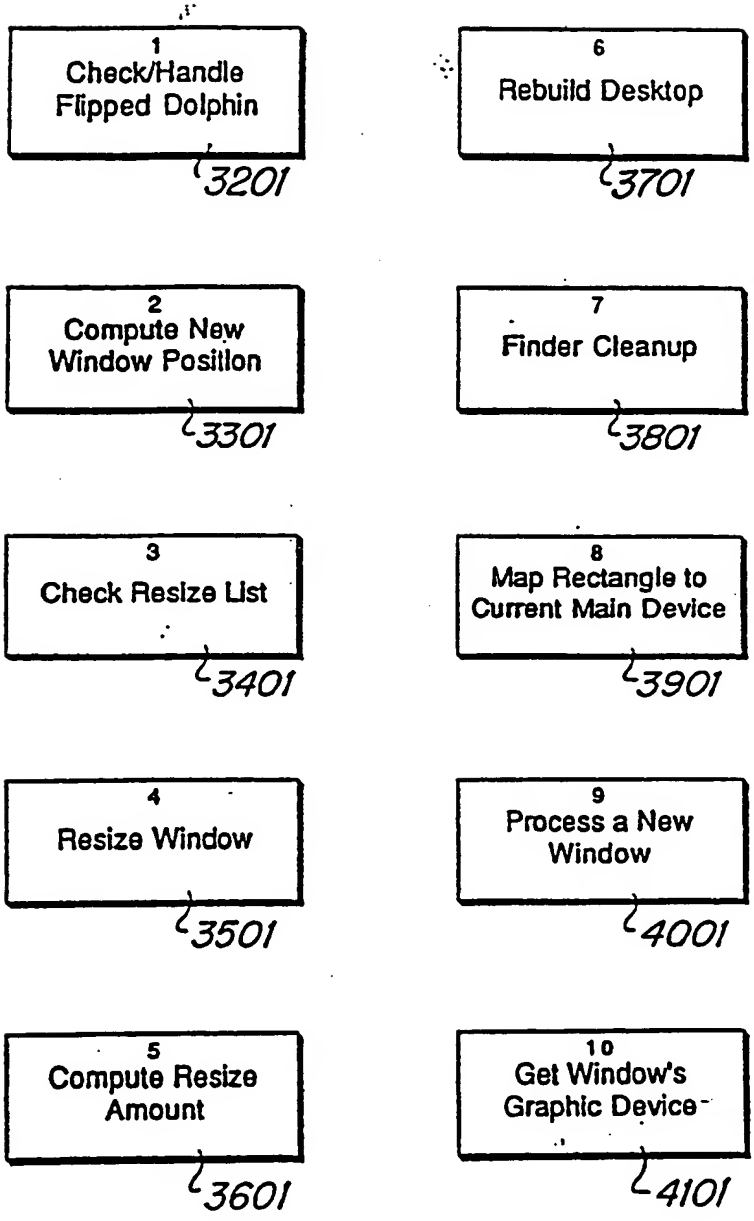
Text Box

117

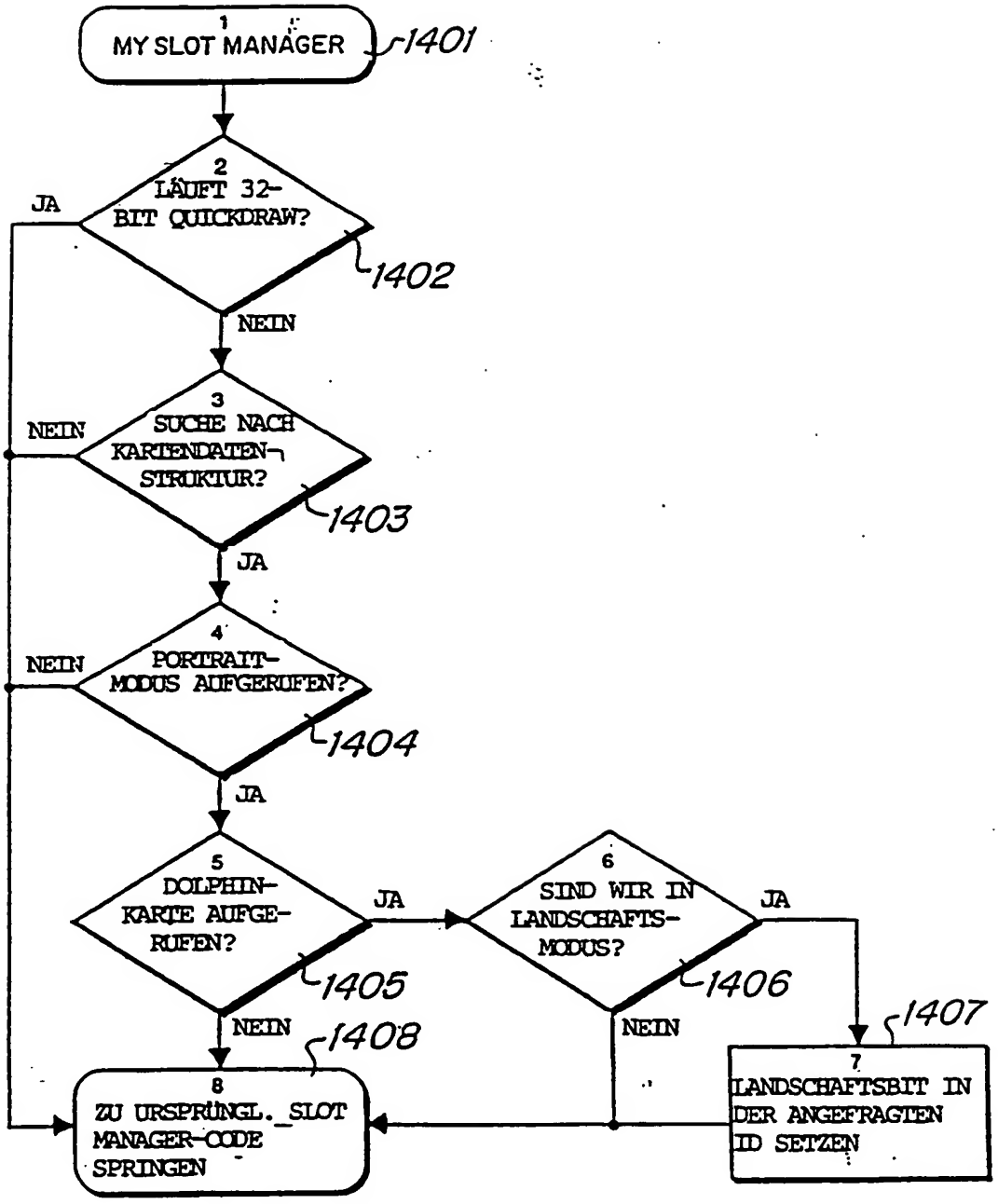
Init Zone

118

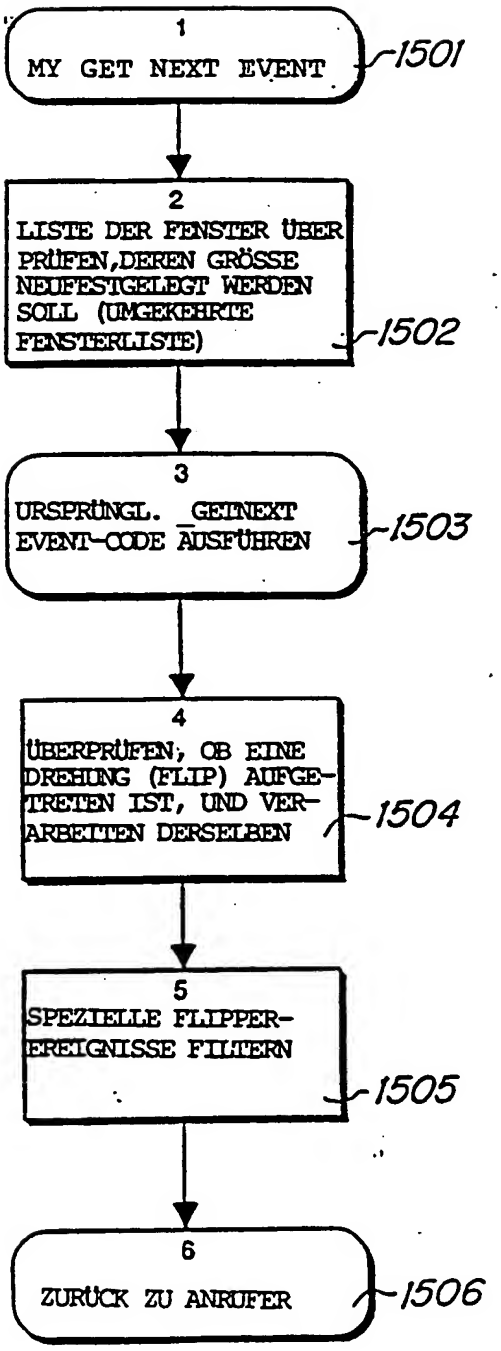
Figur 13a


1
Check/Handle Flipped Dolphin
3201
2
Compute New Window Position
3301
3
Check Resize List
3401
4
Resize Window
3501
5
Compute Resize Amount
3601
6
Rebuild Desktop
3701
7
Finder Cleanup
3801
8
Map Rectangle to Current Main Device
3901
9
Process a New Window
4001
10
Get Window's Graphic Device
4101

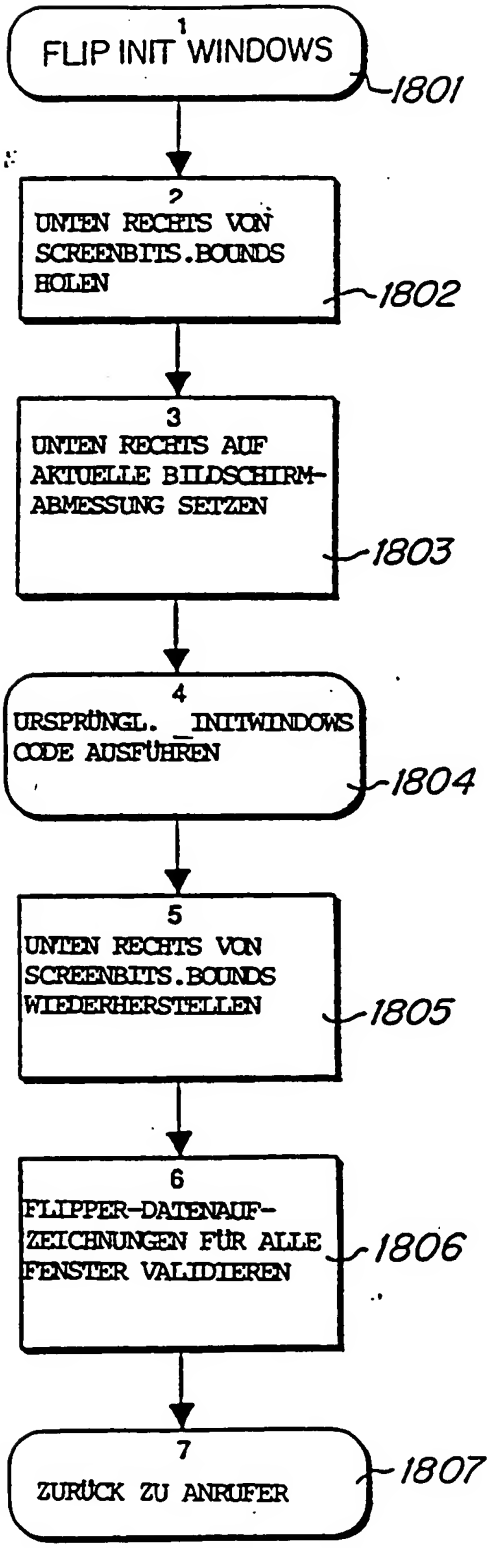
Figur 13b


1
MY SLOT MANAGER
1401
2
LÄUFT 32-BIT QUICKDRAW?
1402
JA
NEIN
3
SUCHE NACH KARTENDATEN-STRUKTUR?
1403
NEIN
JA
4
PORTRAIT-MODUS AUFGERUFEN?
1404
NEIN
JA
5
DOLPHIN-KARTE AUFGERUFEN?
1405
JA
6
SIND WIR IN LANDSCHAFTS-MODUS?
1406
JA
NEIN
1407
7
LANDSCHAFTSBIT IN DER ANGEFRAGTEN ID SETZEN
NEIN
1408
8
ZU URSPRÜNGL._SLOT MANAGER-CODE SPRINGEN

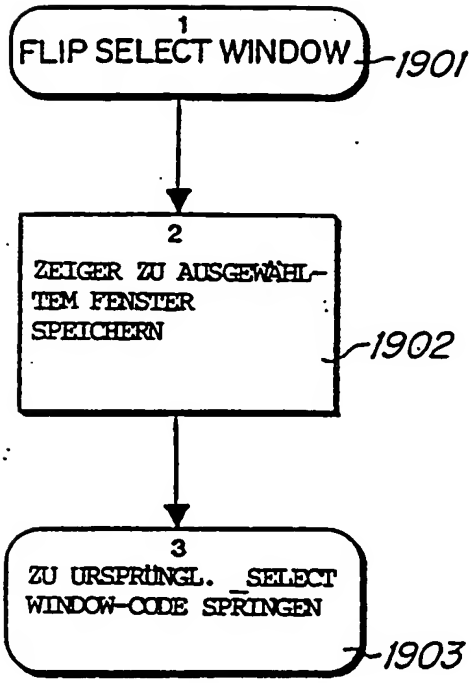
Figur 14


1
MY GET NEXT EVENT
1501
2
LISTE DER FENSTER ÜBER PRÜFEN, DEREN GRÖSSE NEUFESTGELEGT WERDEN SOLL (UMGEKEHRTE FENSTERLISTE)
1502
3
URSPRÜNGL. _GETNEXT EVENT-CODE AUSFÜHREN
1503
4
ÜBERPRÜFEN, OB EINE DREHUNG (FLIP) AUFGETRETEN IST, UND VERARBEITEN DERSELBEN
1504
5
SPEZIELLE FLIPPEREREIGNISSE FILTERN
1505
6
ZURÜCK ZU ANRUFER
1506

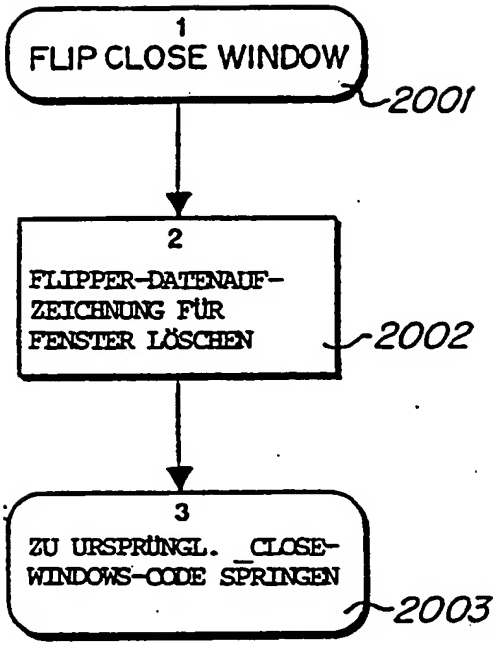
Figur 15


1
FLIP INIT WINDOWS
1801
2
UNTEN RECHTS VON SCREENBITS.BOUNDS HOLEN
1802
3
UNTEN RECHTS AUF AKTUELLE BILDSCHIRM-ABMESSUNG SETZEN
1803
4
URSPRÜNGL. _INITWINDOWS CODE AUSFÜHREN
1804
5
UNTEN RECHTS VON SCREENBITS.BOUNDS WIEDERHERSTELLEN
1805
6
FLIPPER-DATENAUF-ZEICHNUNGEN FÜR ALLE FENSTER VALIDIEREN
1806
7
ZURÜCK ZU ANRUFER
1807

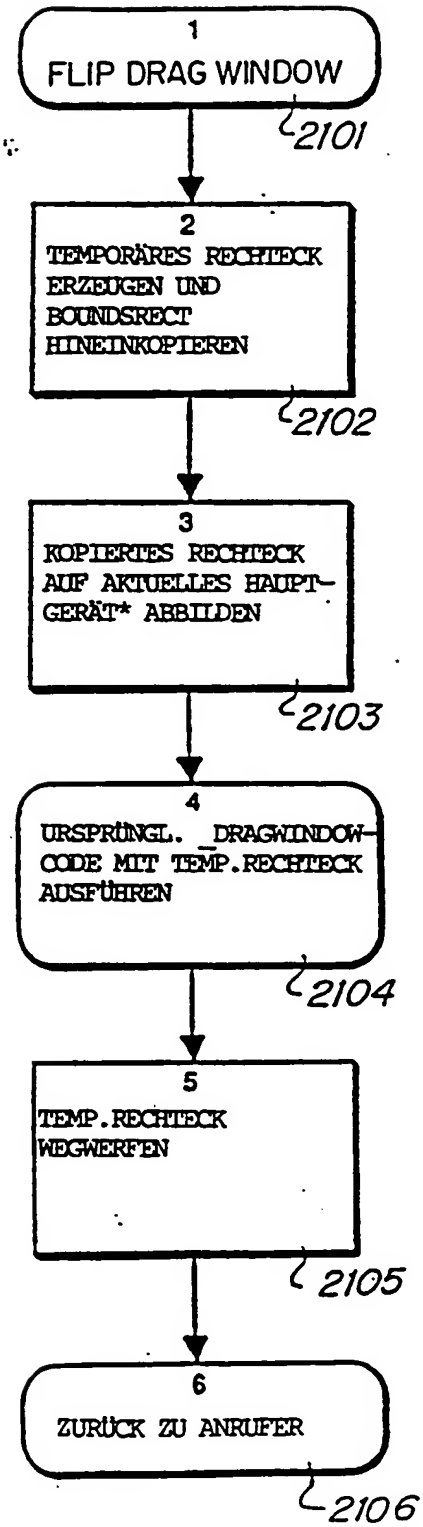
Figur 18


1
FLIP SELECT WINDOW
1901
2
ZEIGER ZU AUSGEWÄHL-
TEM FENSTER
SPEICHERN
1902
3
ZU URSPRÜNGL. _SELECT
WINDOW-CODE SPRINGEN
1903

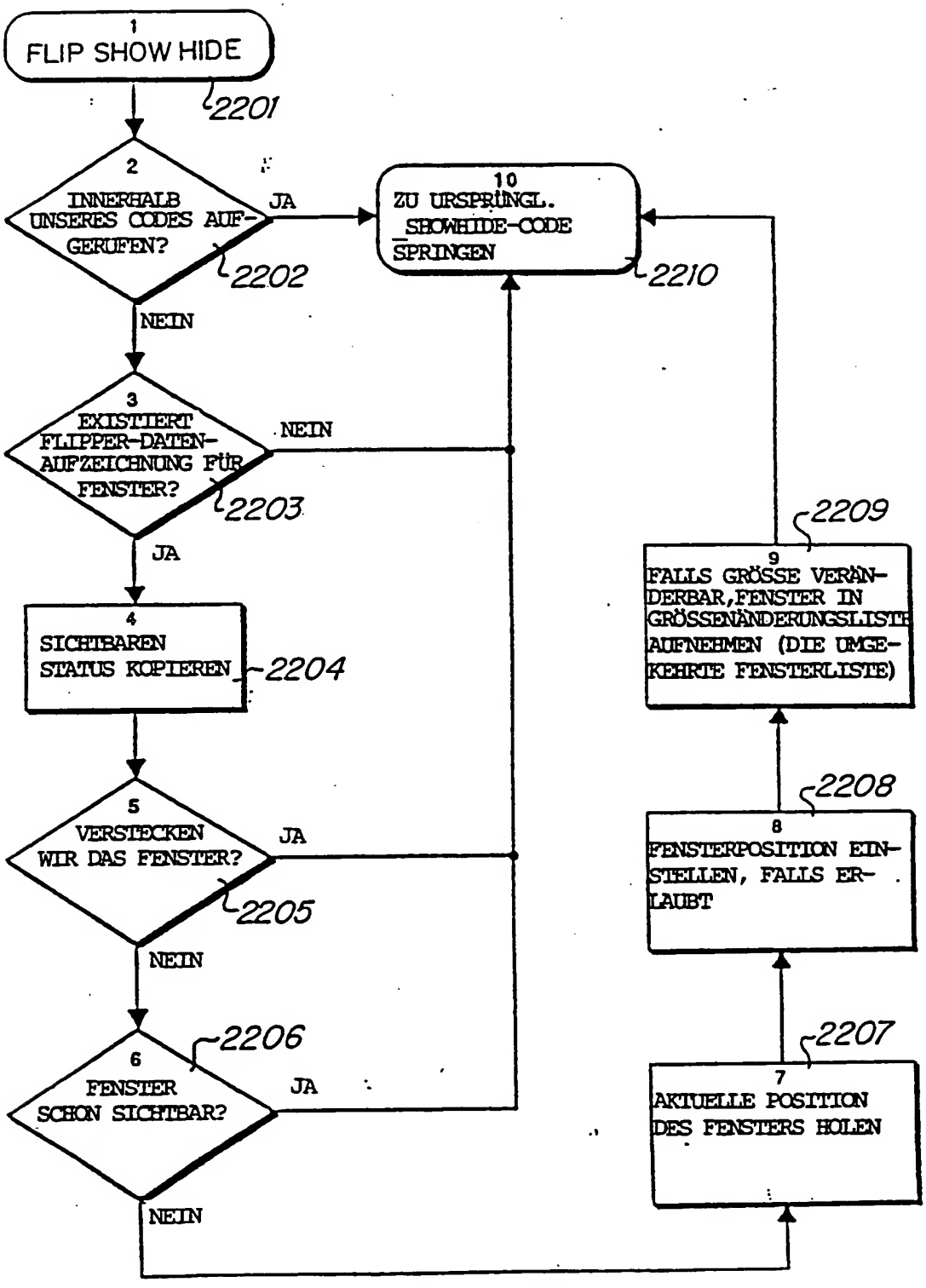
Figur 19


1
FLIP CLOSE WINDOW
2001
2
FLIPPER-DATENAUF-
ZEICHNUNG FÜR
FENSTER LÖSCHEN
2002
3
ZU URSPRÜNGL. _CLOSE-
WINDOWS-CODE SPRINGEN
2003

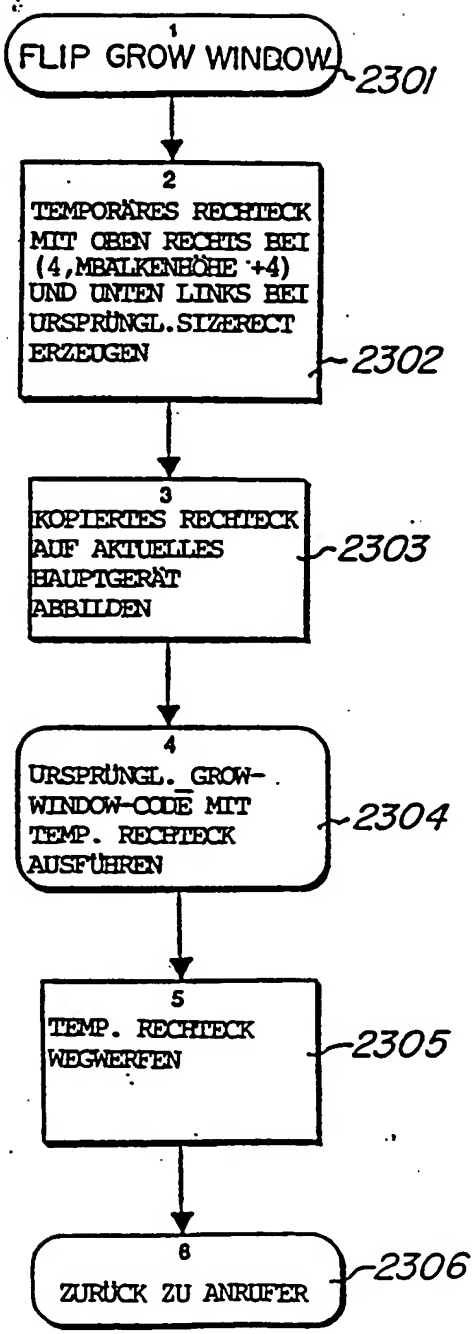
Figur 20


1
FLIP DRAG WINDOW
2101
2
TEMPORÄRES RECHTECK ERZEUGEN UND BOUNDSRECT HINEINKOPIEREN
2102
3
KOPIERTES RECHTECK AUF AKTUELLES HAUPT-GERÄT* ABBILDEN
2103
4
URSPRÜNGL. _DRAGWINDOW-CODE MIT TEMP.RECHTECK AUSFÜHREN
2104
5
TEMP.RECHTECK WEGWERFEN
2105
6
ZURÜCK ZU ANRUFER
2106

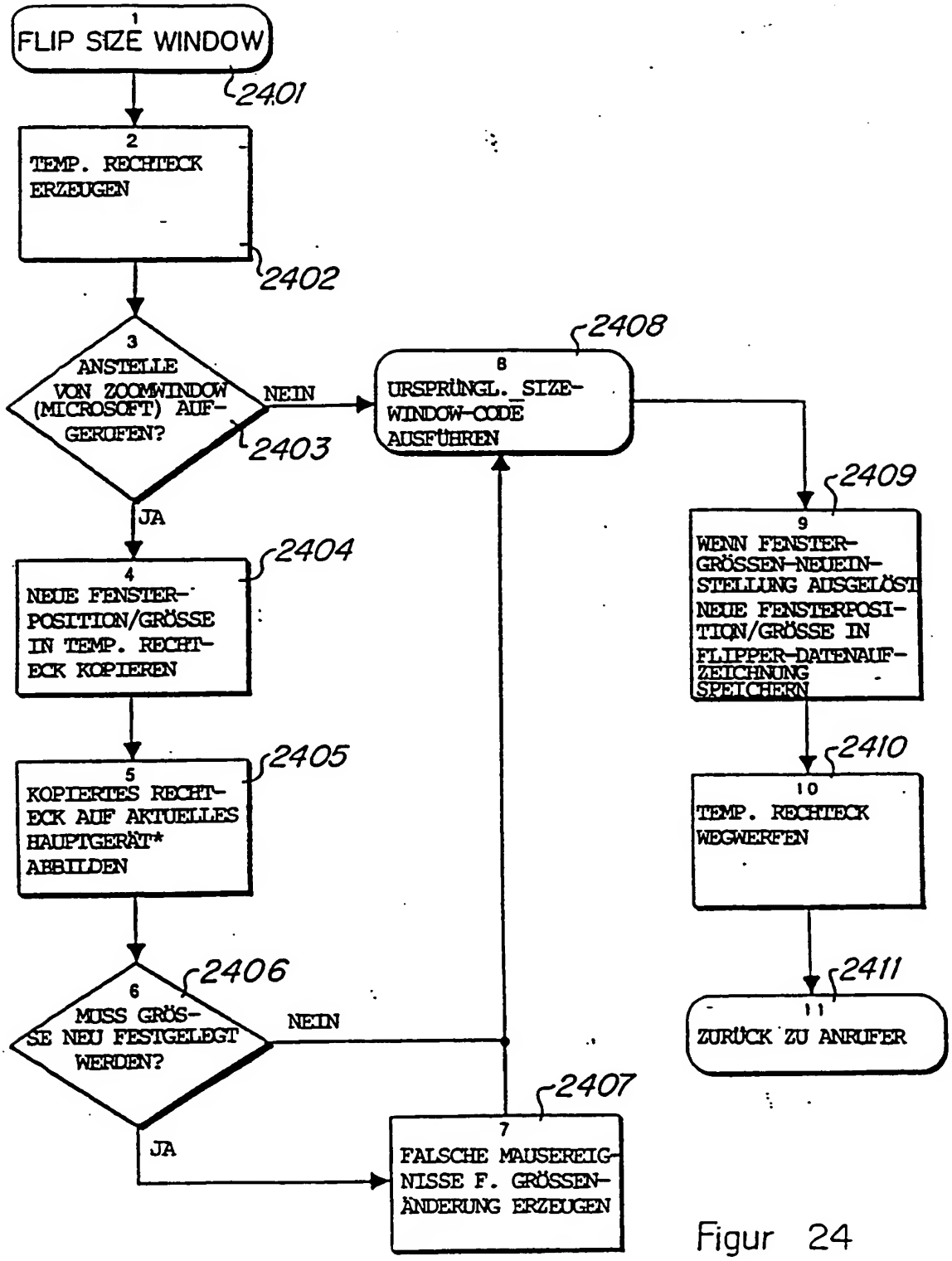
Figur 21


1
FLIP SHOW HIDE
2201
2
INNERHALB UNSERES CODES AUFGERUFEN?
JA
2202
10
ZU URSPRÜNGL. SHOWHIDE-CODE SPRINGEN
2210
NEIN
3
EXISTIERT FLIPPER-DATEN-AUFZEICHNUNG FÜR FENSTER?
NEIN
2203
JA
4
SICHTBAREN STATUS KOPIEREN
2204
5
VERSTECKEN WIR DAS FENSTER?
JA
2205
NEIN
6
FENSTER SCHON SICHTBAR?
2206
JA
NEIN
2209
9
FALLS GRÖSSE VERÄNDERBAR, FENSTER IN GRÖSSENÄNDERUNGSLISTE AUFNEHMEN (DIE UMGEKEHRTE FENSTERLISTE)
2208
8
FENSTERPOSITION EINSTELLEN, FALLS ERLAUBT
2207
7
AKTUELLE POSITION DES FENSTERS HOLEN

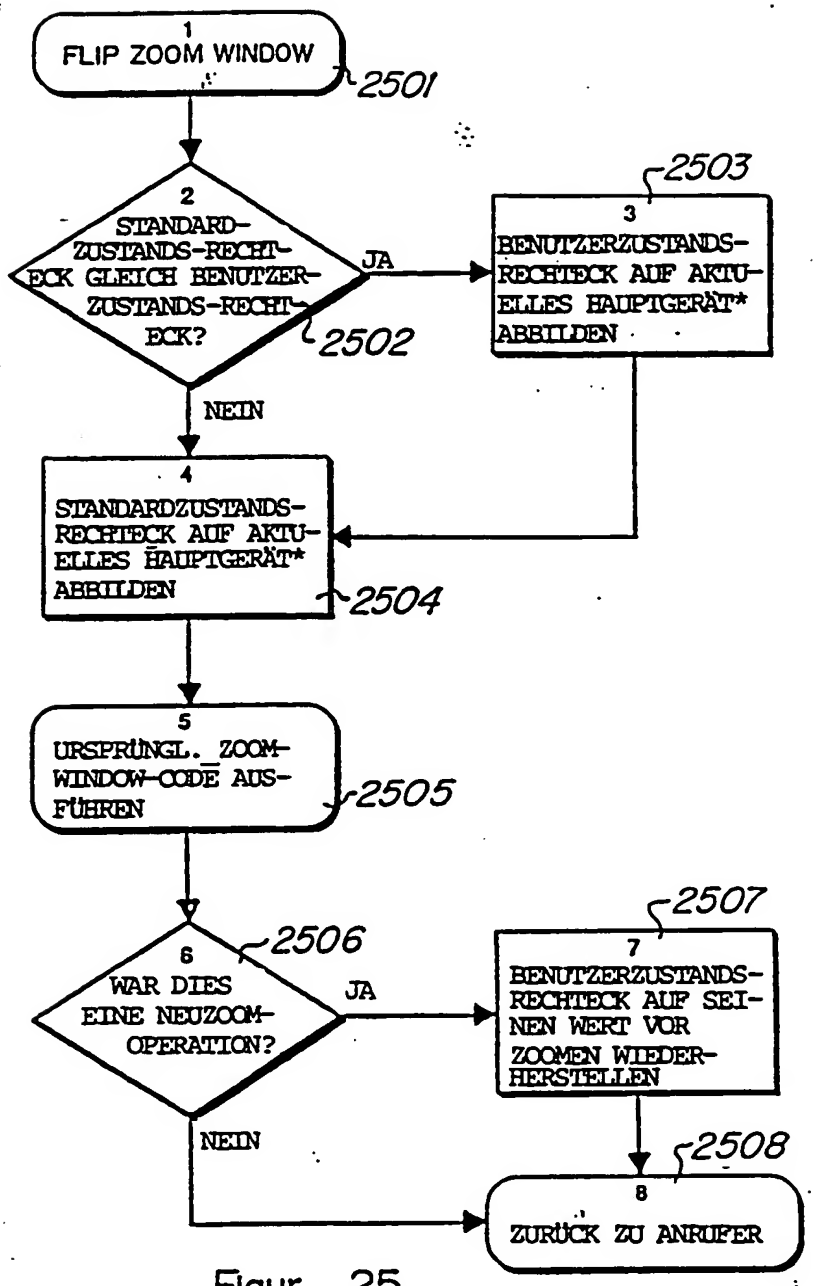
Figur 22


1
FLIP GROW WINDOW
2301
2
TEMPORÄRES RECHTECK MIT OBEN RECHTS BEI (4,MBALKENHÖHE +4) UND UNTEN LINKS BEI URSPRÜNGL.SIZERECT ERZEUGEN
2302
3
KOPIERTES RECHTECK AUF AKTUELLES HAUPTGERÄT ABBILDEN
2303
4
URSPRÜNGL. GROW-WINDOW-CODE MIT TEMP. RECHTECK AUSFÜHREN
2304
5
TEMP. RECHTECK WEGWERFEN
2305
6
ZURÜCK ZU ANRUFER
2306

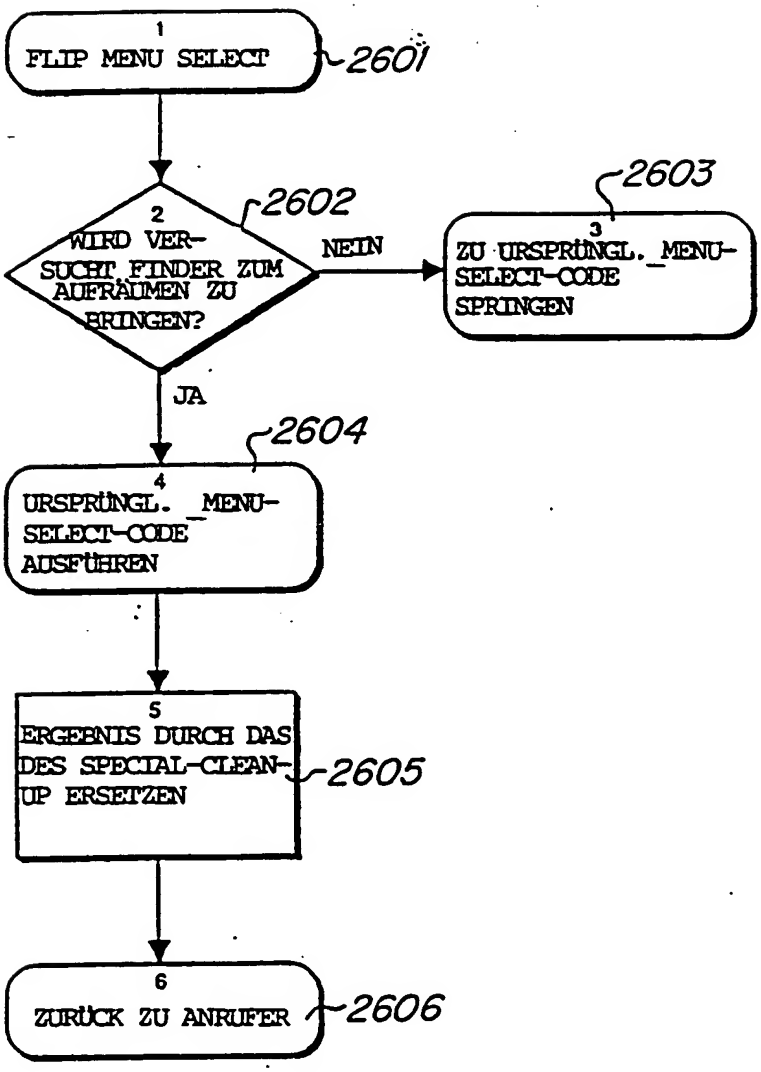
Figur 23


1
FLIP SIZE WINDOW
2401
2
TEMP. RECHTECK ERZEUGEN
2402
3
ANSTELLE VON ZOOMWINDOW (MICROSOFT) AUFGERUFEN?
NEIN
2403
JA
4
NEUE FENSTERPOSITION/GRÖSSE IN TEMP. RECHTECK KOPIEREN
2404
5
KOPIERTES RECHTECK AUF AKTUELLES HAUPTGERÄT* ABBILDEN
2405
6
MUSS GRÖSSE NEU FESTGELEGT WERDEN?
2406
NEIN
JA
7
FALSCHE MAUSEREIGNISSE F. GRÖSSENÄNDERUNG ERZEUGEN
2407
8
URSPRÜNGL._SIZE-WINDOW-CODE AUSFÜHREN
2408
9
WENN FENSTERGRÖSSEN-NEUEINSTELLUNG AUSGELÖST NEUE FENSTERPOSITION/GRÖSSE IN FLIPPER-DATENAUFZEICHNUNG SPEICHERN
2409
10
TEMP. RECHTECK WEGWERFEN
2410
11
ZURÜCK ZU ANRUFER
2411

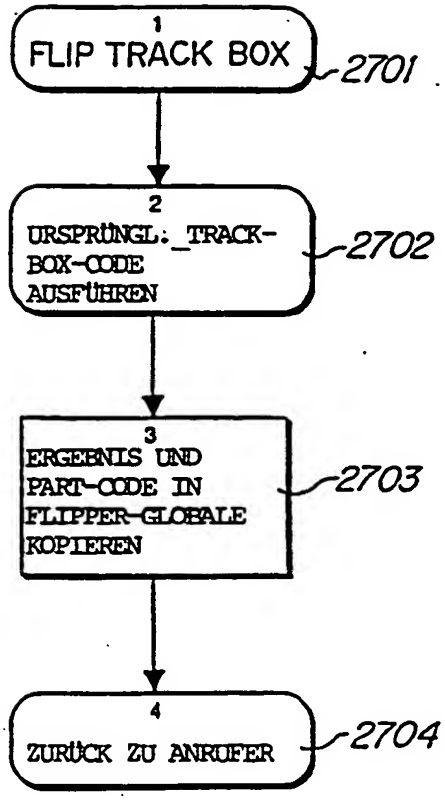
Figur 24


1
FLIP ZOOM WINDOW
2501
2
STANDARD-ZUSTANDS-RECHTECK GLEICH BENUTZER-ZUSTANDS-RECHTECK?
2502
JA
2503
3
BENUTZERZUSTANDS-RECHTECK AUF AKTUELLES HAUPTGERÄT* ABBILDEN
NEIN
4
STANDARDZUSTANDS-RECHTECK AUF AKTUELLES HAUPTGERÄT* ABBILDEN
2504
5
URSPRÜNGL. ZOOM-WINDOW-CODE AUSFÜHREN
2505
6
WAR DIES EINE NEUZOOM-OPERATION?
2506
JA
2507
7
BENUTZERZUSTANDS-RECHTECK AUF SEINEN WERT VOR ZOOMEN WIEDERHERSTELLEN
NEIN
2508
8
ZURÜCK ZU ANRUFER

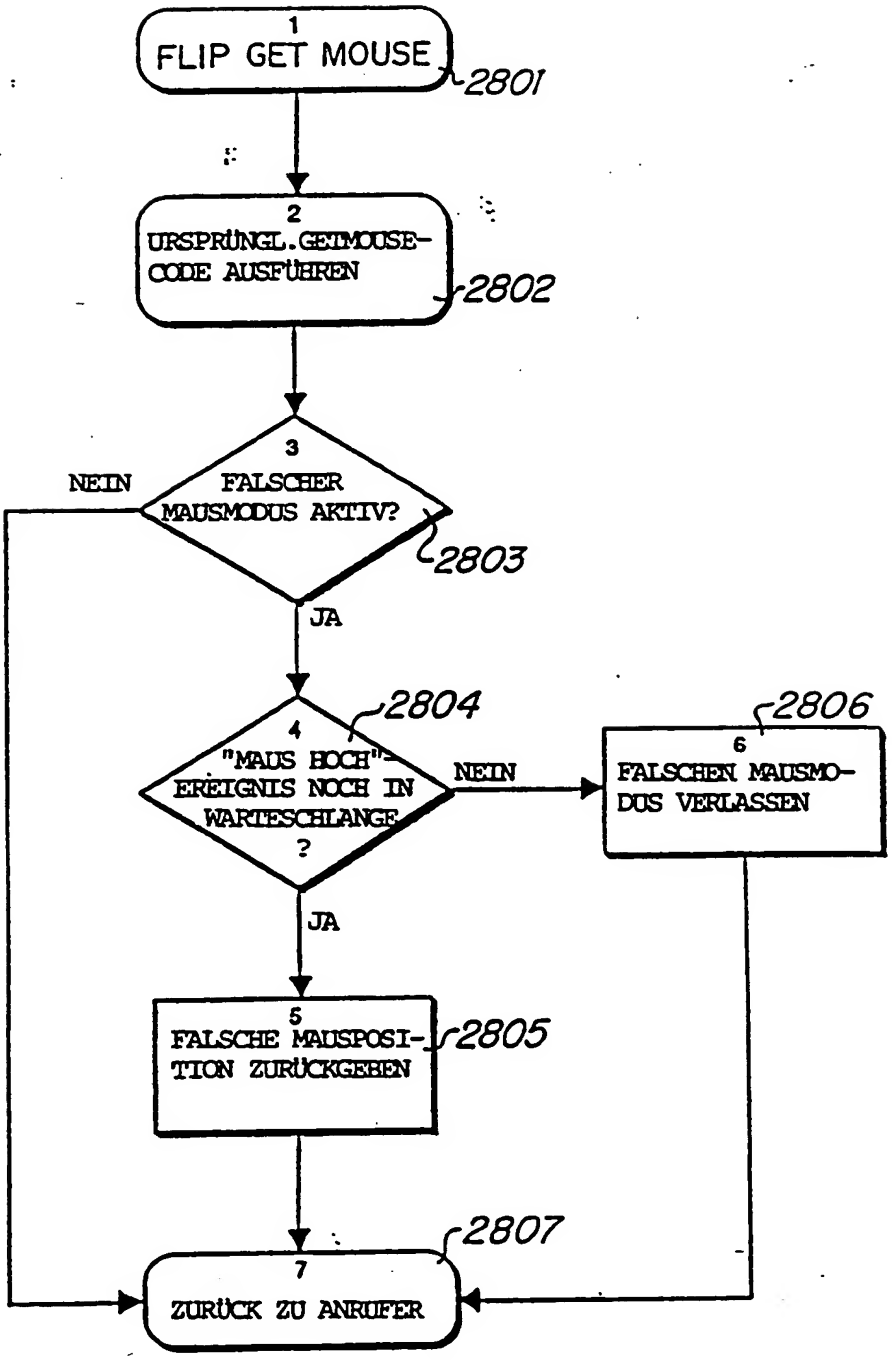
Figur 25


1
FLIP MENU SELECT
2601
2
WIRD VER-
SUCHT FINDER ZUM
AUFRÄUMEN ZU
BRINGEN?
2602
NEIN
3
ZU URSPRÜNGL._MENU-
SELECT-CODE
SPRINGEN
2603
JA
4
URSPRÜNGL._MENU-
SELECT-CODE
AUSFÜHREN
2604
5
ERGEBNIS DURCH DAS
DES SPECIAL-CLEAN-
UP ERSETZEN
2605
6
ZURÜCK ZU ANRUFER
2606

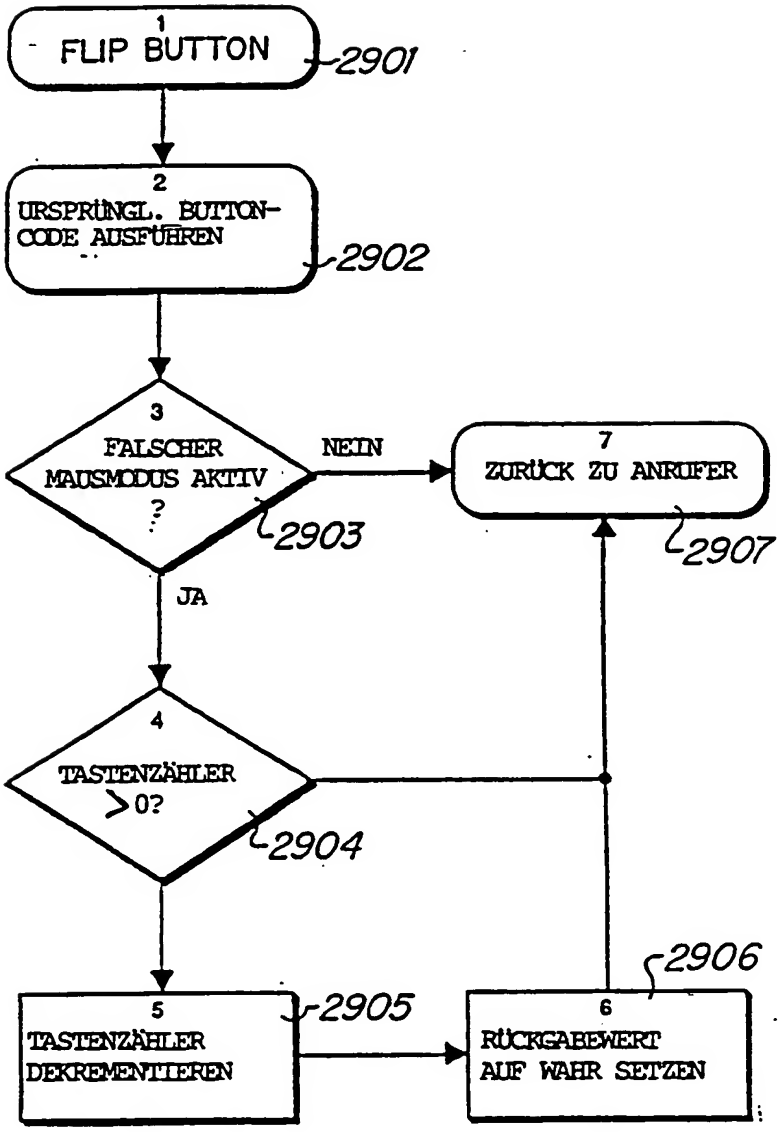
Figur 26


1
FLIP TRACK BOX
2701
2
URSPRÜNGL._TRACK-
BOX-CODE
AUSFÜHREN
2702
3
ERGEBNIS UND
PART-CODE IN
FLIPPER-GLOBALE
KOPIEREN
2703
4
ZURÜCK ZU ANRUFER
2704

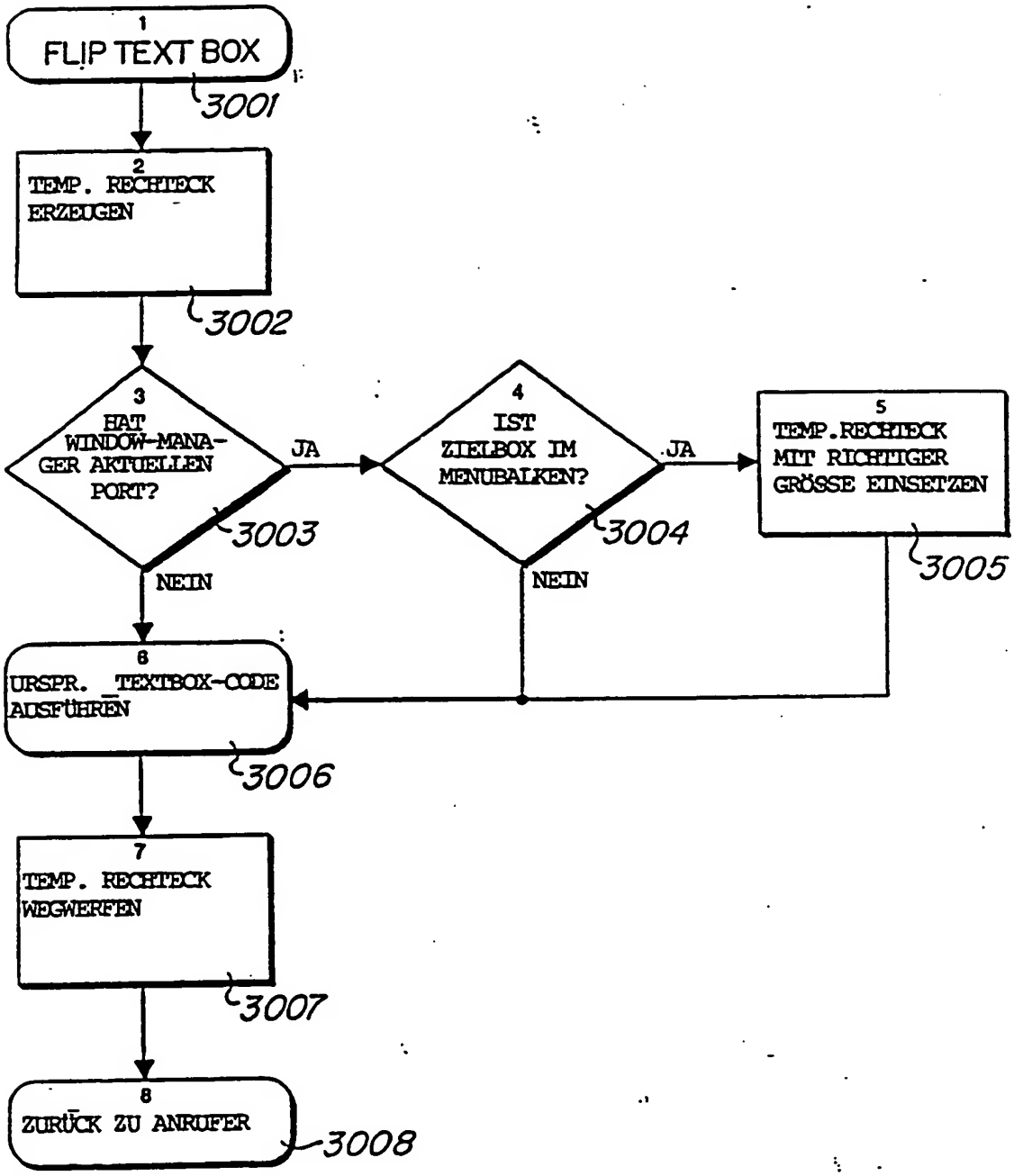
Figur 27


1
FLIP GET MOUSE
2801
2
URSPRÜNGL.GETMOUSE-CODE AUSFÜHREN
2802
3
FALSCHER MAUSMODUS AKTIV?
2803
NEIN
JA
2804
4
"MAUS HOCH"-EREIGNIS NOCH IN WARTESCHLANGE ?
NEIN
2806
6
FALSCHEN MAUSMODUS VERLASSEN
JA
5
FALSCHE MAUSPOSITION ZURÜCKGEBEN
2805
2807
7
ZURÜCK ZU ANRUFER

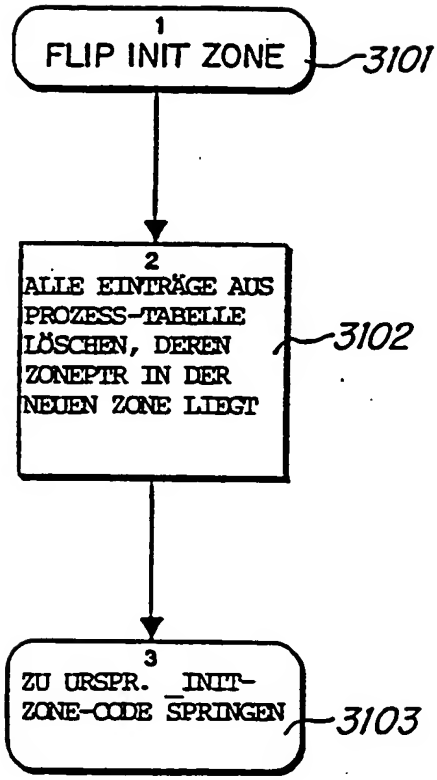
Figur 28


1
FLIP BUTTON
2901
2
URSPRÜNGL. BUTTON-CODE AUSFÜHREN
2902
3
FALSCHER MAUSMODUS AKTIV ?
2903
NEIN
7
ZURÜCK ZU ANRUFER
2907
JA
4
TASTENZÄHLER > 0?
2904
2906
5
TASTENZÄHLER DEKREMENTIEREN
2905
6
RÜCKGABEWERT AUF WAHR SETZEN

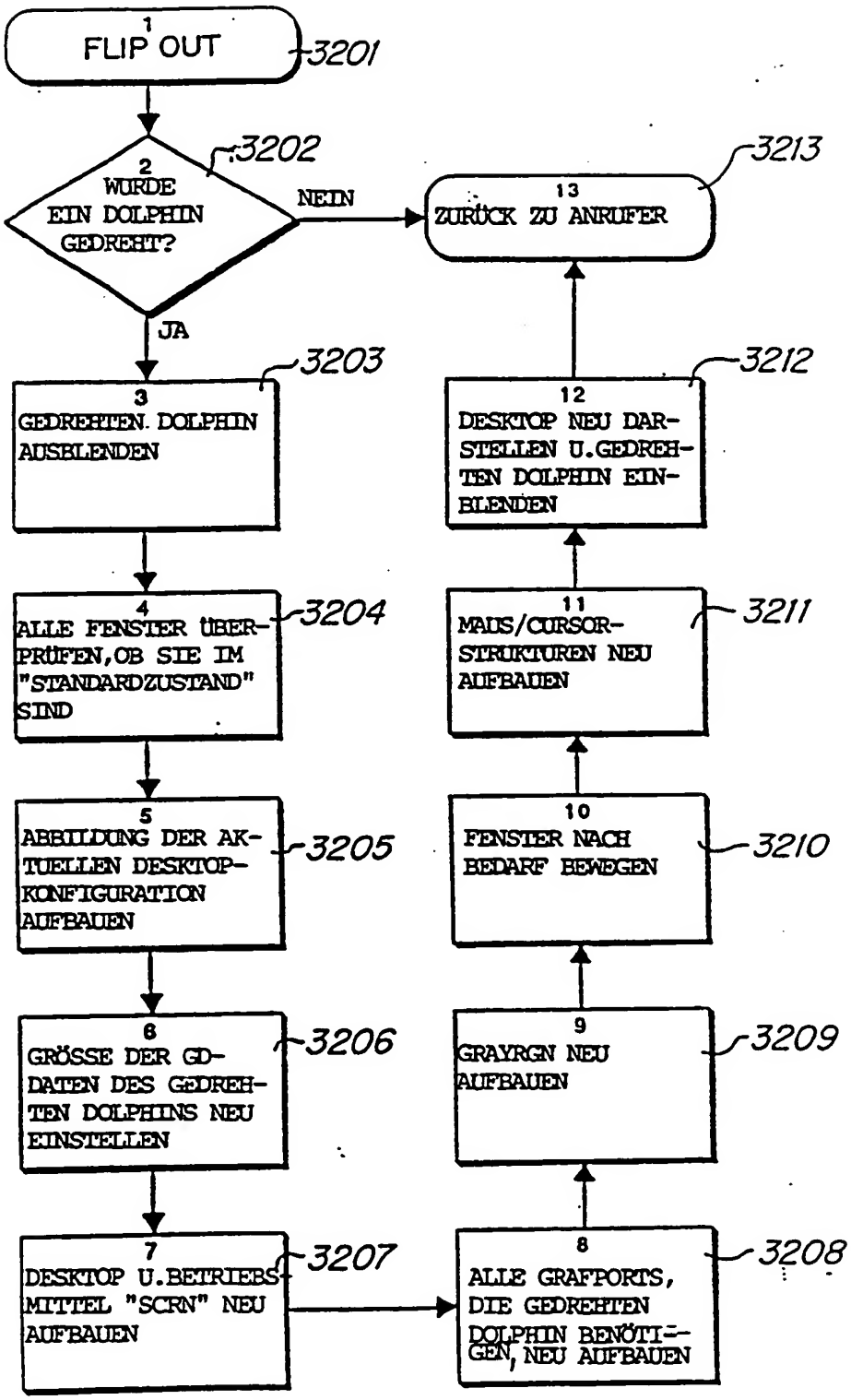
Figur 29


1
FLIP TEXT BOX
3001
2
TEMP. RECHTECK ERZEUGEN
3002
3
HAT WINDOW-MANAGER AKTUELLEN PORT?
JA
3003
NEIN
4
IST ZIELBOX IM MENUBALKEN?
JA
3004
NEIN
5
TEMP.RECHTECK MIT RICHTIGER GRÖSSE EINSETZEN
3005
6
URSPR. _TEXTBOX-CODE AUSFÜHREN
3006
7
TEMP. RECHTECK WEGWERFEN
3007
8
ZURÜCK ZU ANRUFER
3008

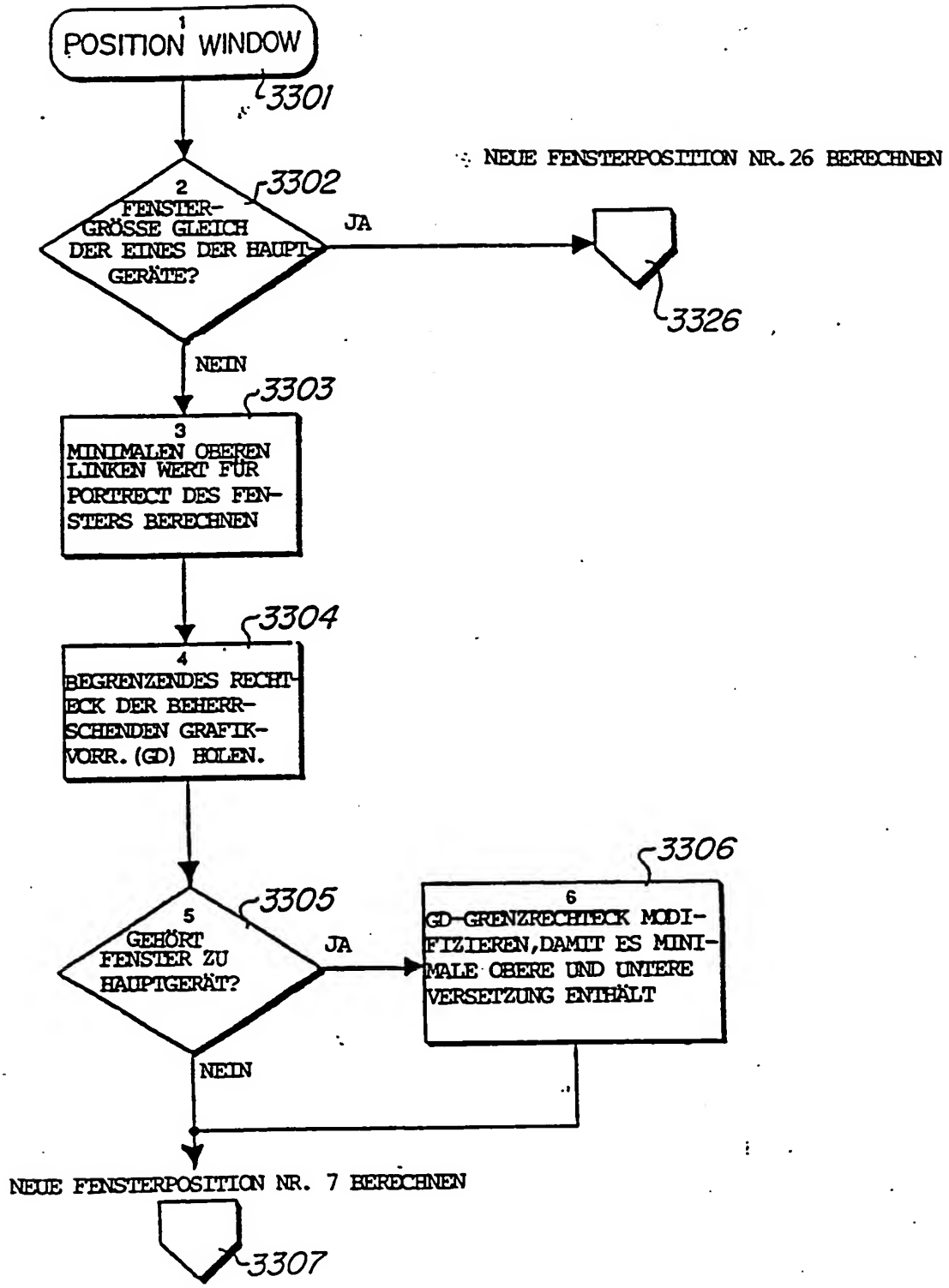
Figur 30


1
FLIP INIT ZONE
3101
2
ALLE EINTRÄGE AUS PROZESS-TABELLE LÖSCHEN, DEREN ZONEPTR IN DER NEUEN ZONE LIEGT
3102
3
ZU URSPR. _INIT-ZONE-CODE SPRINGEN
3103

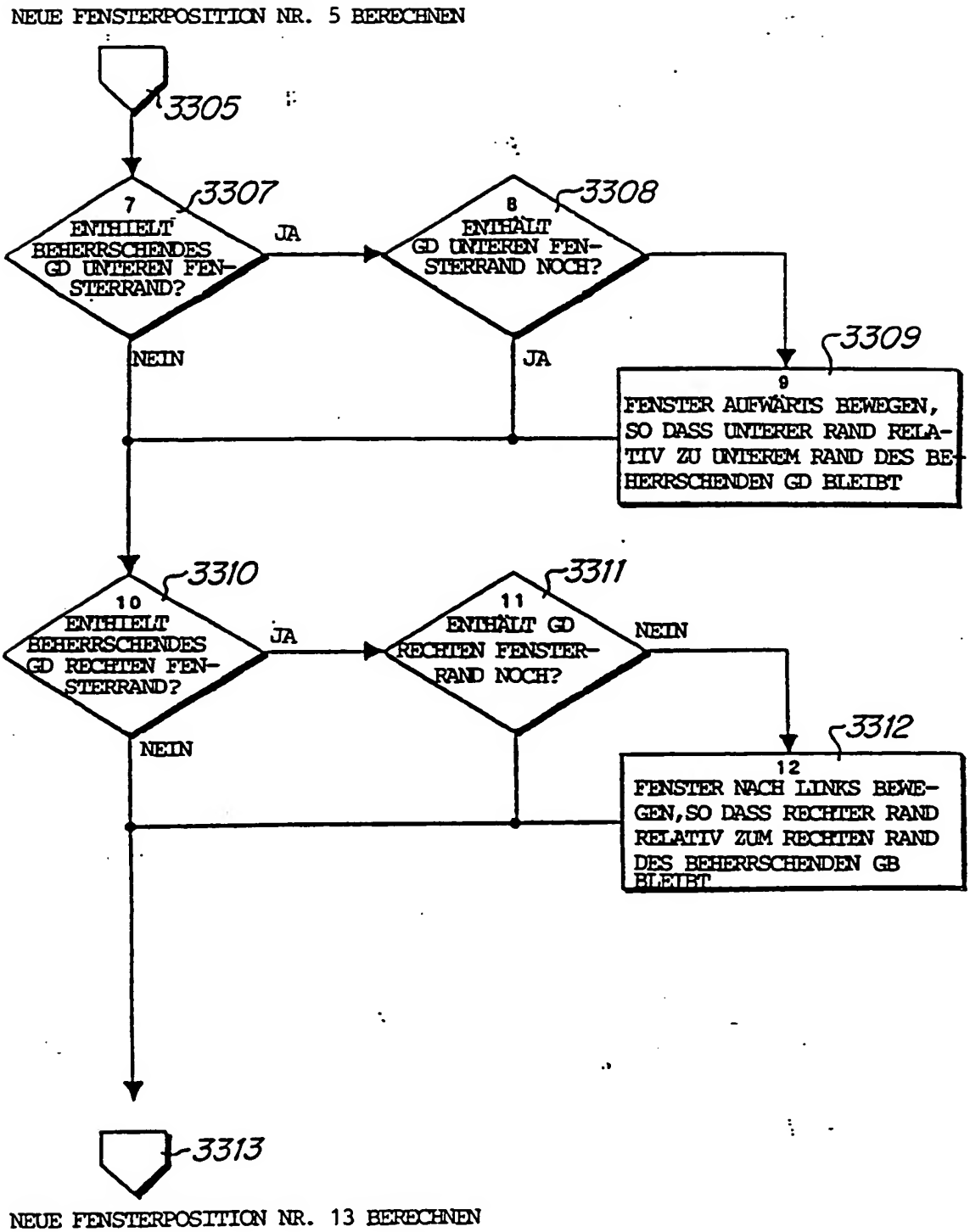
Figur 31


1
FLIP OUT
3201
2
WURDE EIN DOLPHIN GEDREHT?
3202
NEIN
JA
13
ZURÜCK ZU ANRUFER
3213
3
GEDREHTEN DOLPHIN AUSBLENDEN
3203
12
DESKTOP NEU DARSTELLEN U.GEDREHTEN DOLPHIN EINBLENDEN
3212
4
ALLE FENSTER ÜBERPRÜFEN,OB SIE IM "STANDARDZUSTAND" SIND
3204
11
MAUS/CURSORSTRUKTUREN NEU AUFBAUEN
3211
5
ABBILDUNG DER AKTUELLEN DESKTOPKONFIGURATION AUFBAUEN
3205
10
FENSTER NACH BEDARF BEWEGEN
3210
6
GRÖSSE DER GDDATEN DES GEDREHTEN DOLPHINS NEU EINSTELLEN
3206
9
GRAYRGN NEU AUFBAUEN
3209
7
DESKTOP U.BETRIEBSMITTEL "SCRN" NEU AUFBAUEN
3207
8
ALLE GRAFPORTS, DIE GEDREHTEN DOLPHIN BENÖTIGEN, NEU AUFBAUEN
3208

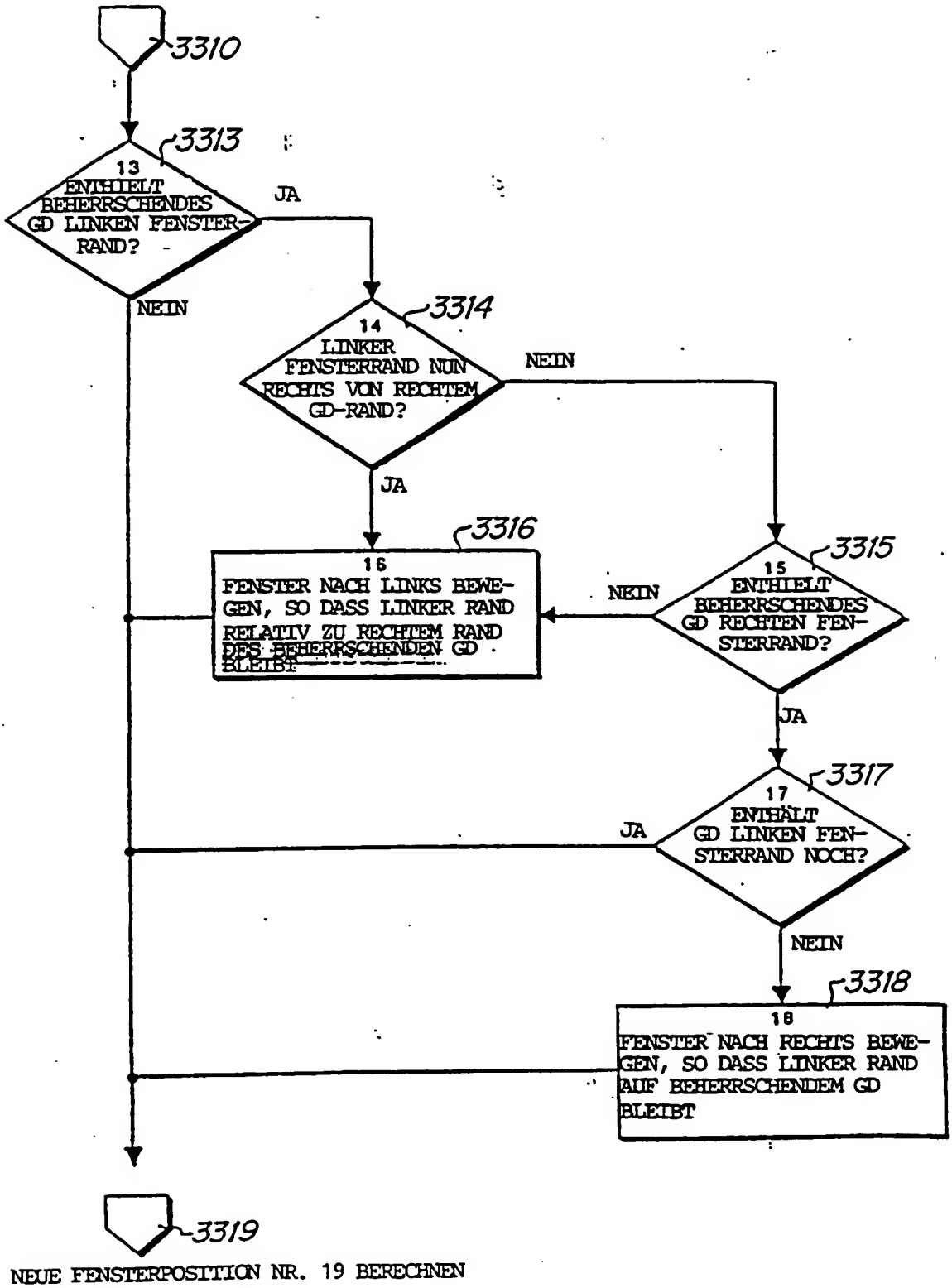
Figur 32


1
POSITION WINDOW
3301
2
FENSTER-GRÖSSE GLEICH DER EINES DER HAUPT-GERÄTE?
3302
JA
NEUE FENSTERPOSITION NR. 26 BERECHNEN
3326
NEIN
3
MINIMALEN OBEREN LINKEN WERT FÜR PORTRECT DES FENSTERS BERECHNEN
3303
4
BEGRENZENDES RECHTECK DER BEHERRSCHENDEN GRAFIKVORR. (GD) HOLEN.
3304
5
GEHÖRT FENSTER ZU HAUPTGERÄT?
3305
JA
6
GD-GRENZRECHTECK MODIFIZIEREN, DAMIT ES MINIMALE OBERE UND UNTERE VERSETZUNG ENTHÄLT
3306
NEIN
NEUE FENSTERPOSITION NR. 7 BERECHNEN
3307

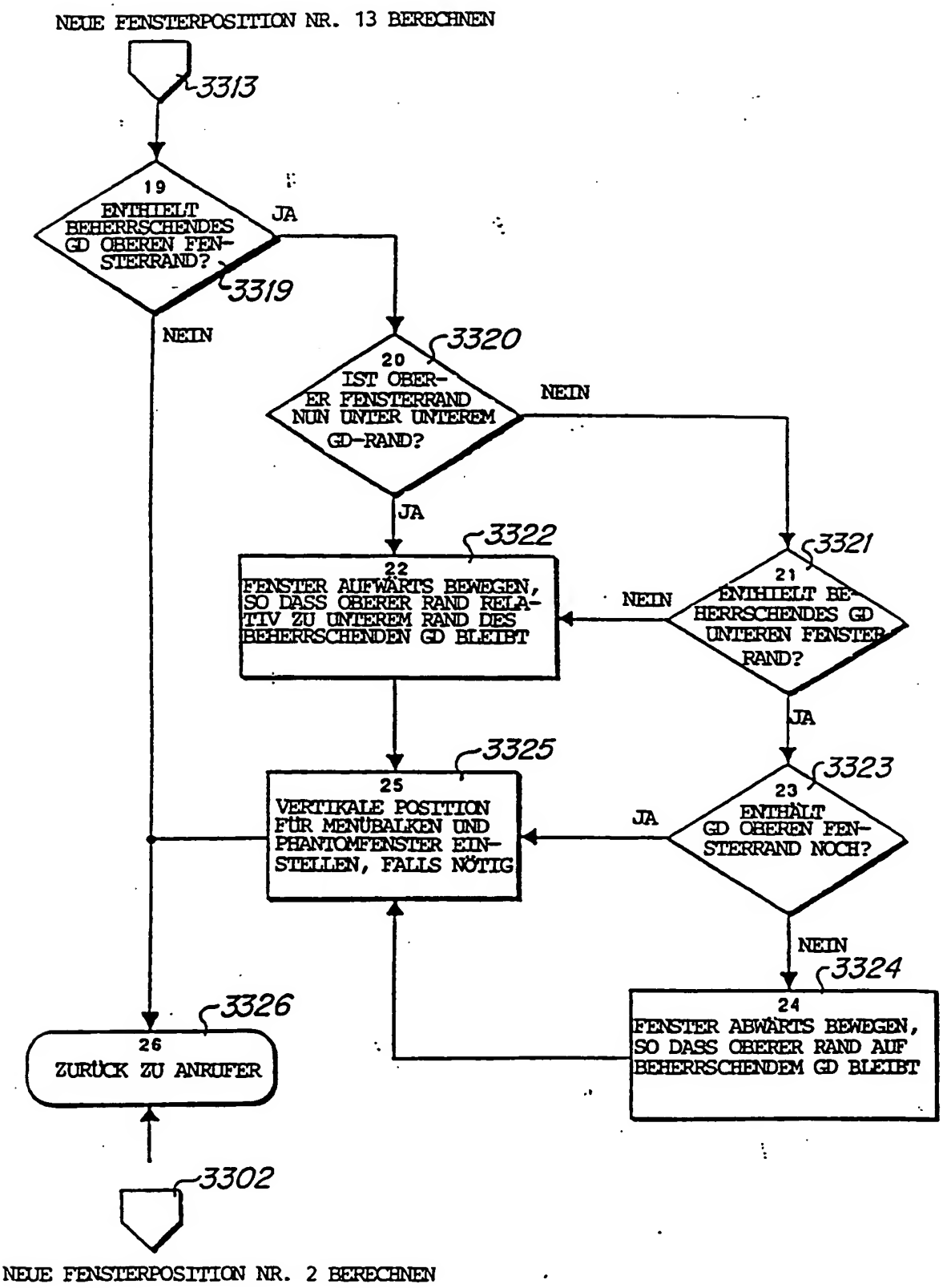
Figur. 33a


NEUE FENSTERPOSITION NR. 5 BERECHNEN
3305
7
ENTHIELT BEHERRSCHENDES GD UNTEREN FENSTERRAND?
3307
JA
8
ENTHÄLT GD UNTEREN FENSTERRAND NOCH?
3308
NEIN
JA
9
FENSTER AUFWÄRTS BEWEGEN, SO DASS UNTERER RAND RELATIV ZU UNTEREM RAND DES BEHERRSCHENDEN GD BLEIBT
3309
10
ENTHIELT BEHERRSCHENDES GD RECHTEN FENSTERRAND?
3310
JA
11
ENTHÄLT GD RECHTEN FENSTERRAND NOCH?
3311
NEIN
NEIN
12
FENSTER NACH LINKS BEWEGEN, SO DASS RECHTER RAND RELATIV ZUM RECHTEN RAND DES BEHERRSCHENDEN GB BLEIBT
3312
3313
NEUE FENSTERPOSITION NR. 13 BERECHNEN

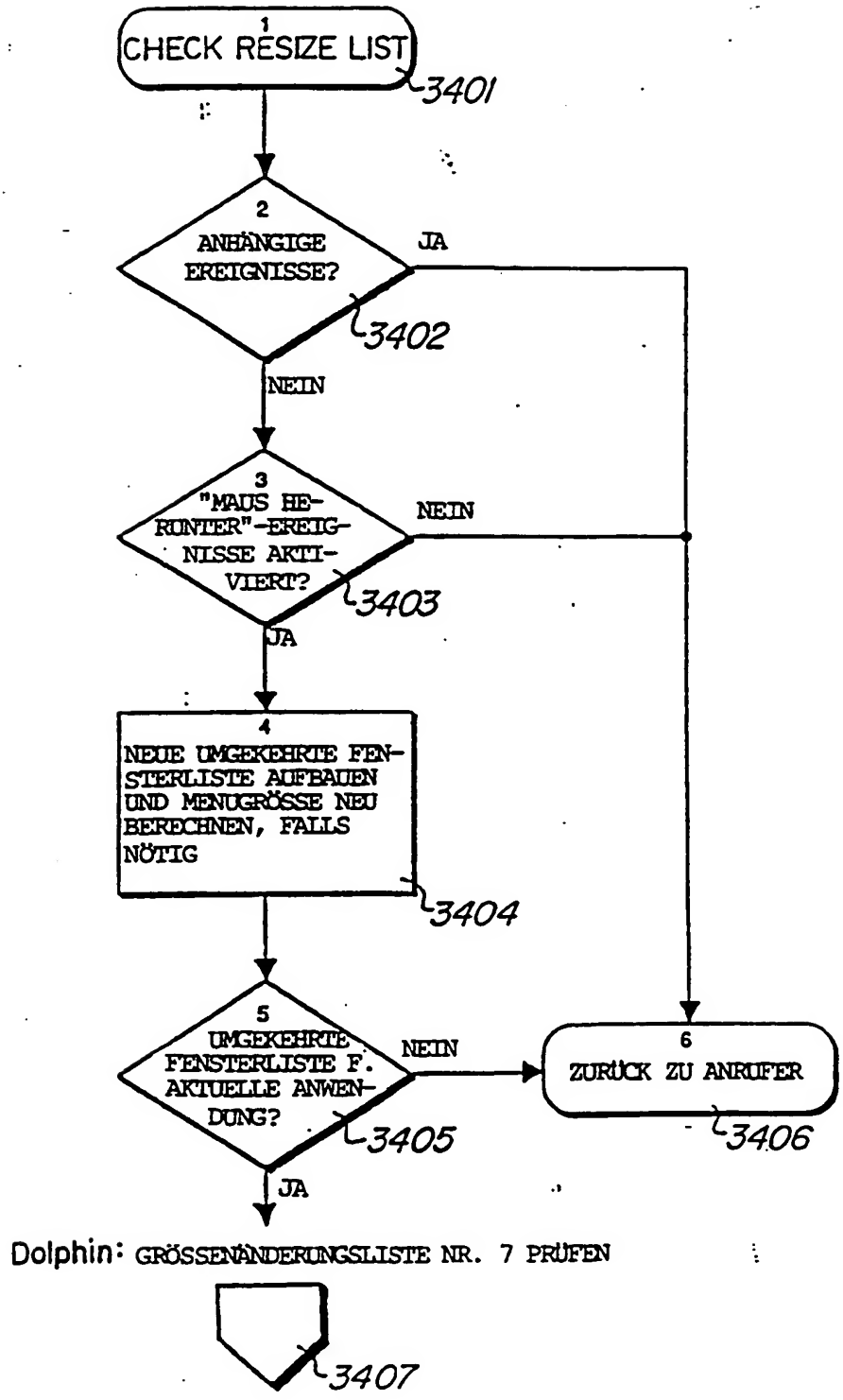
Figur 33b


NEUE FENSTERPOSITION NR. 10 BERECHNEN
3310
3313
13
ENTHIELT BEHERRSCHENDES GD LINKEN FENSTER-RAND?
JA
NEIN
3314
14
LINKER FENSTERRAND NUN RECHTS VON RECHTEM GD-RAND?
NEIN
JA
3316
16
FENSTER NACH LINKS BEWEGEN, SO DASS LINKER RAND RELATIV ZU RECHTEM RAND DES BEHERRSCHENDEN GD BLEIBT
NEIN
3315
15
ENTHIELT BEHERRSCHENDES GD RECHTEN FENSTERRAND?
JA
3317
17
ENTHÄLT GD LINKEN FENSTERRAND NOCH?
JA
NEIN
3318
18
FENSTER NACH RECHTS BEWEGEN, SO DASS LINKER RAND AUF BEHERRSCHENDEM GD BLEIBT
3319
NEUE FENSTERPOSITION NR. 19 BERECHNEN

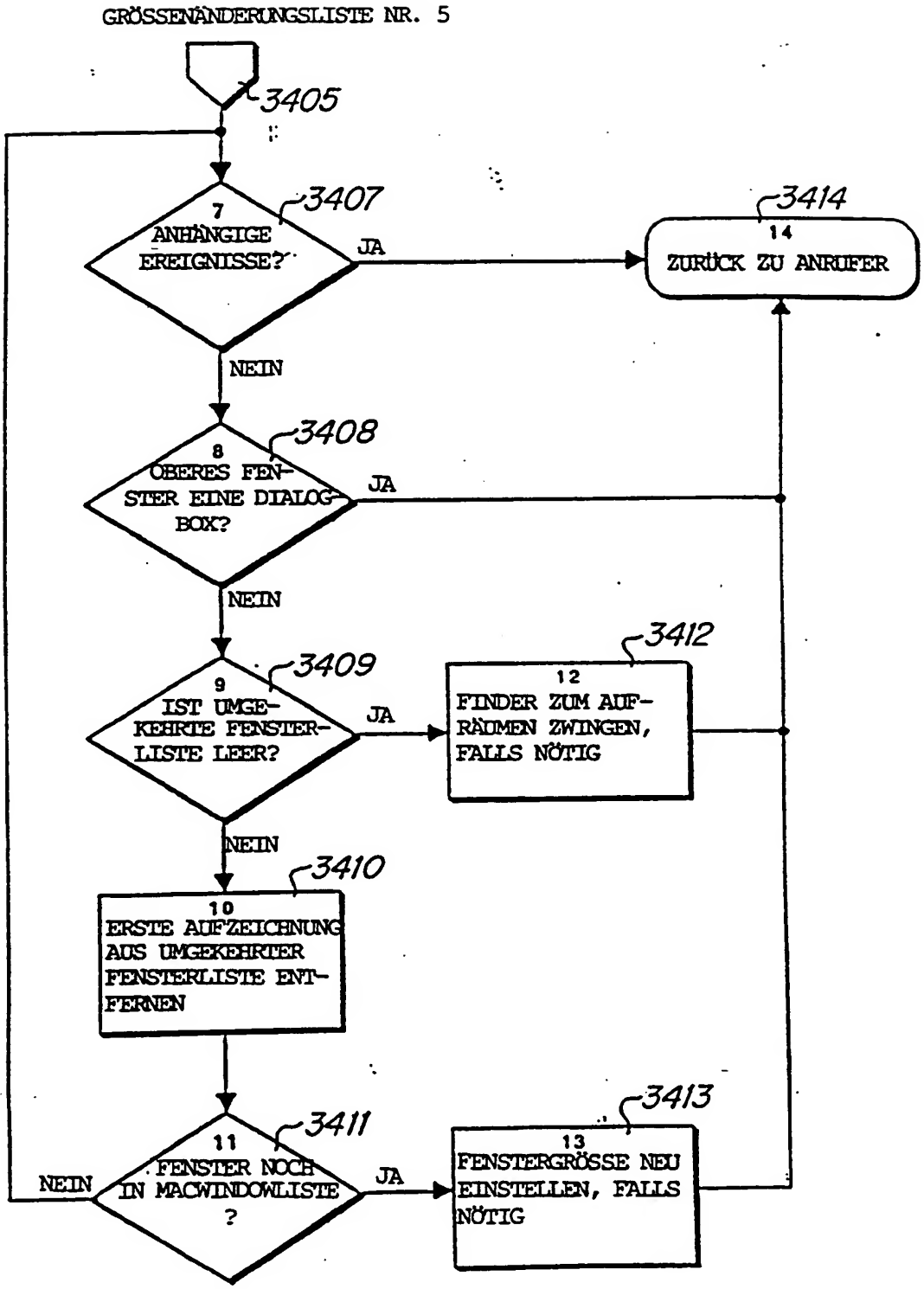
Figur 33c


NEUE FENSTERPOSITION NR. 13 BERECHNEN
3313
19
ENTHIELT BEHERRSCHENDES GD OBEREN FENSTERRAND?
JA
3319
NEIN
3320
20
IST OBERER FENSTERRAND NUN UNTER UNTEREM GD-RAND?
NEIN
JA
3322
22
FENSTER AUFWÄRTS BEWEGEN, SO DASS OBERER RAND RELATIV ZU UNTEREM RAND DES BEHERRSCHENDEN GD BLEIBT
NEIN
3321
21
ENTHIELT BEHERRSCHENDES GD UNTEREN FENSTERRAND?
JA
3325
25
VERTIKALE POSITION FÜR MENÜBALKEN UND PHANTOMFENSTER EINSTELLEN, FALLS NÖTIG
JA
3323
23
ENTHÄLT GD OBEREN FENSTERRAND NOCH?
NEIN
3324
24
FENSTER ABWÄRTS BEWEGEN, SO DASS OBERER RAND AUF BEHERRSCHENDEM GD BLEIBT
3326
26
ZURÜCK ZU ANRUFER
3302
NEUE FENSTERPOSITION NR. 2 BERECHNEN

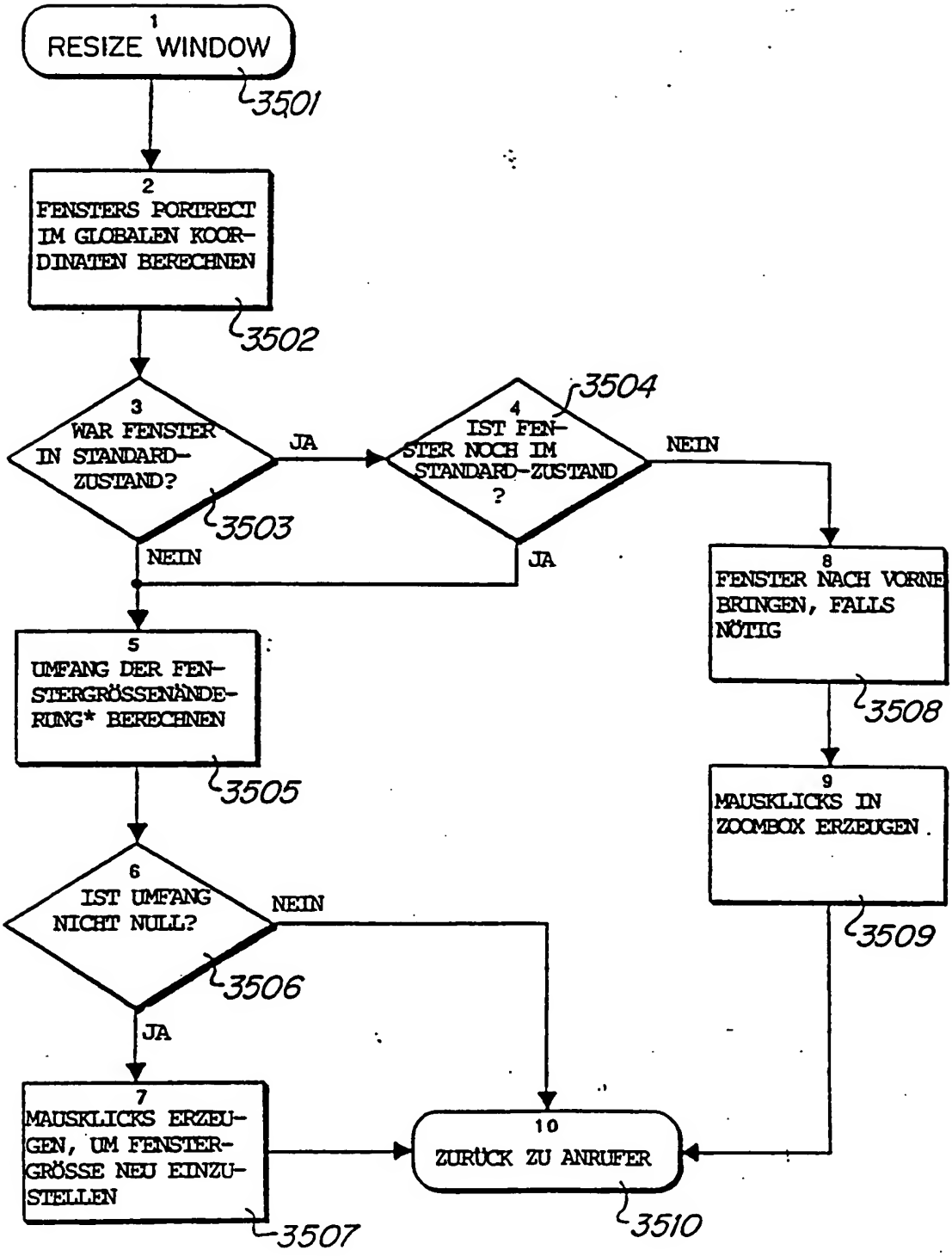
Figur 33d


1
CHECK RESIZE LIST
3401
2
ANHÄNGIGE EREIGNISSE?
JA
3402
NEIN
3
"MAUS HERUNTER"-EREIGNISSE AKTIVIERT?
NEIN
3403
JA
4
NEUE UMGEKEHRTE FENSTERLISTE AUFBAUEN UND MENUGRÖSSE NEU BERECHNEN, FALLS NÖTIG
3404
5
UMGEKEHRTE FENSTERLISTE F. AKTUELLE ANWENDUNG?
NEIN
3405
6
ZURÜCK ZU ANRUFER
3406
JA
Dolphin: GRÖSSENÄNDERUNGSLISTE NR. 7 PRÜFEN
3407

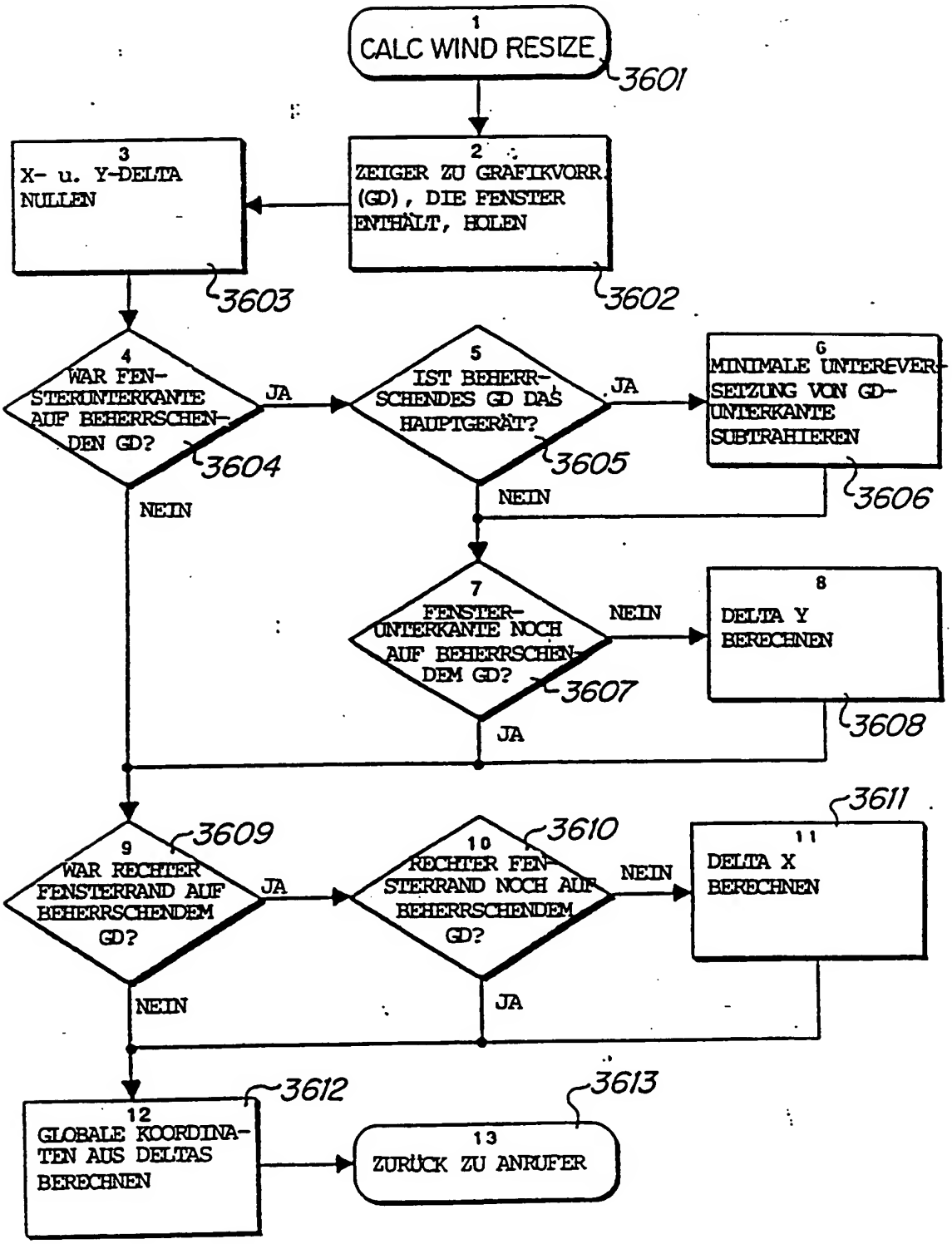
Figur 34a


GRÖSSENÄNDERUNGSLISTE NR. 5
3405
7
ANHÄNGIGE EREIGNISSE?
3407
JA
3414
14
ZURÜCK ZU ANRUFER
NEIN
8
ÖBERES FENSTER EINE DIALOG-BOX?
3408
JA
NEIN
9
IST UMGEKEHRTE FENSTERLISTE LEER?
3409
JA
3412
12
FINDER ZUM AUFRÄUMEN ZWINGEN, FALLS NÖTIG
NEIN
10
ERSTE AUFZEICHNUNG AUS UMGEKEHRTER FENSTERLISTE ENTFERNEN
3410
11
FENSTER NOCH IN MACWINDOWLISTE ?
3411
NEIN
JA
3413
13
FENSTERGRÖSSE NEU EINSTELLEN, FALLS NÖTIG

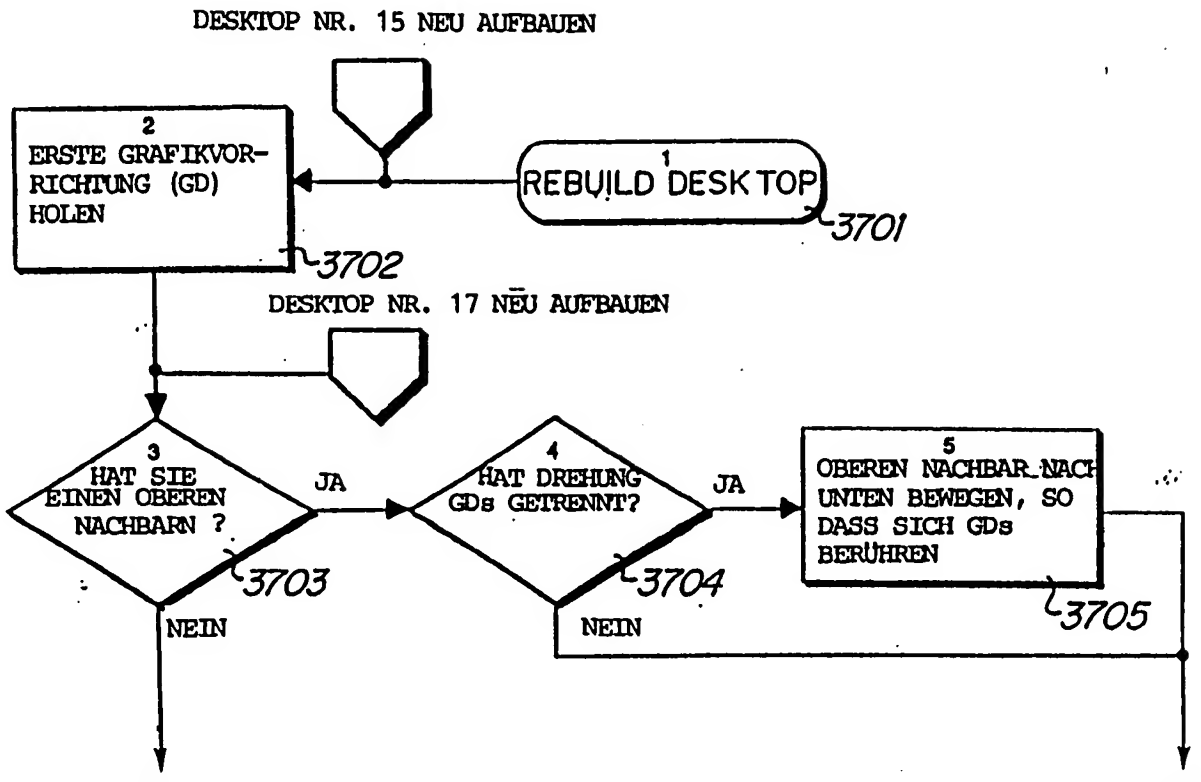
Figur 34b


1
RESIZE WINDOW
3501
2
FENSTERS PORTRECT IM GLOBALEN KOORDINATEN BERECHNEN
3502
3
WAR FENSTER IN STANDARD-ZUSTAND?
JA
3503
NEIN
4
IST FENSTER NOCH IM STANDARD-ZUSTAND ?
3504
NEIN
JA
5
UMFANG DER FENSTERGRÖSSENÄNDERUNG* BERECHNEN
3505
6
IST UMFANG NICHT NULL?
NEIN
3506
JA
7
MAUSKLICKS ERZEUGEN, UM FENSTERGRÖSSE NEU EINZUSTELLEN
3507
8
FENSTER NACH VORNE BRINGEN, FALLS NÖTIG
3508
9
MAUSKLICKS IN ZOOMBOX ERZEUGEN.
3509
10
ZURÜCK ZU ANRUFER
3510

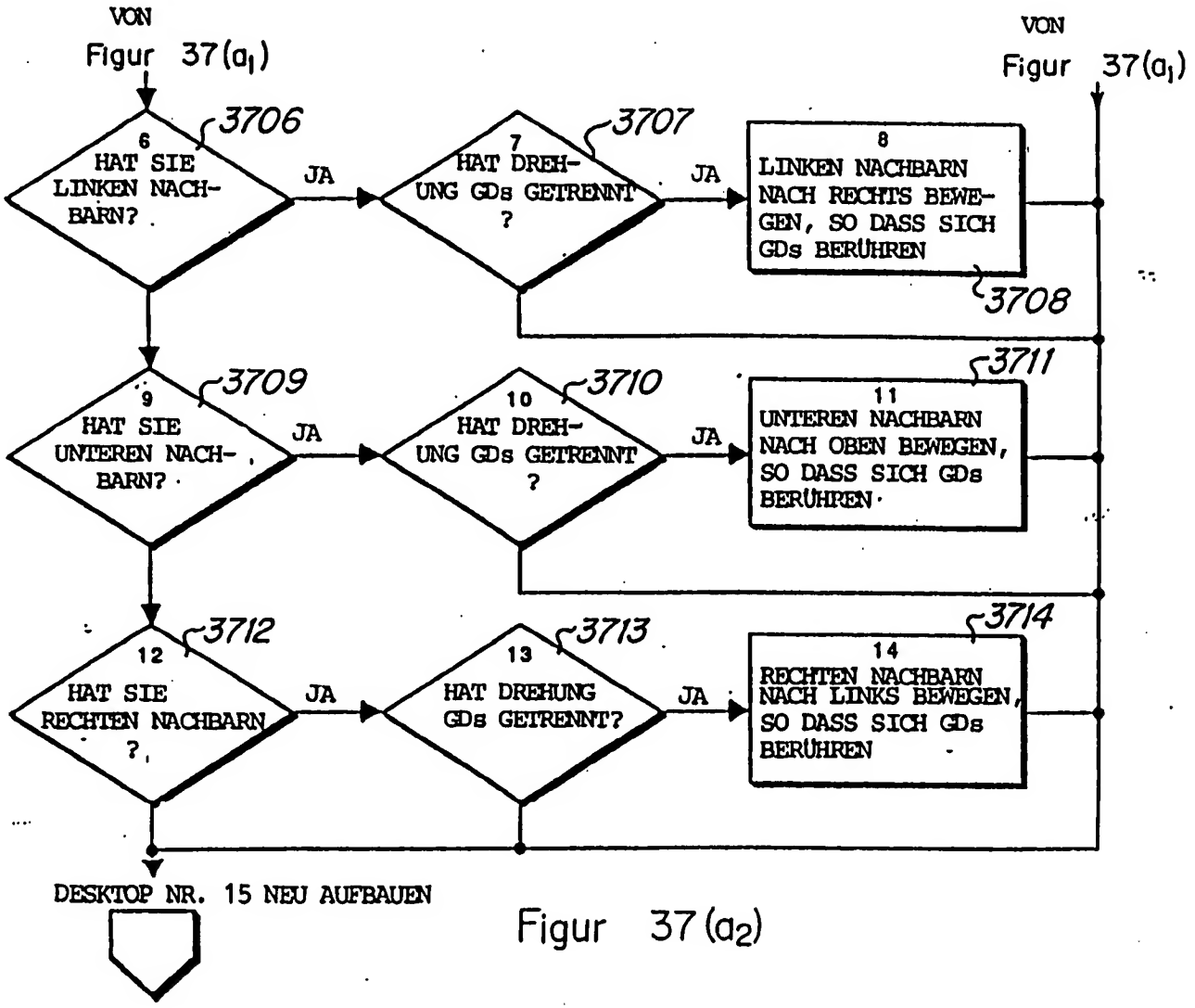
Figur 35


1
CALC WIND RESIZE
3601
2
ZEIGER ZU GRAFIKVORR. (GD), DIE FENSTER ENTHÄLT, HOLEN
3602
3
X- u. Y-DELTA NULLEN
3603
4
WAR FENSTERUNTERKANTE AUF BEHERRSCHENDEN GD?
3604
JA
5
IST BEHERRSCHENDES GD DAS HAUPTGERÄT?
3605
JA
6
MINIMALE UNTEREVERSETZUNG VON GD-UNTERKANTE SUBTRAHIEREN
3606
NEIN
NEIN
7
FENSTERUNTERKANTE NOCH AUF BEHERRSCHENDEM GD?
3607
NEIN
8
DELTA Y BERECHNEN
3608
JA
3609
9
WAR RECHTER FENSTERRAND AUF BEHERRSCHENDEM GD?
JA
3610
10
RECHTER FENSTERRAND NOCH AUF BEHERRSCHENDEM GD?
NEIN
3611
11
DELTA X BERECHNEN
NEIN
JA
3612
12
GLOBALE KOORDINATEN AUS DELTAS BERECHNEN
3613
13
ZURÜCK ZU ANRUFER

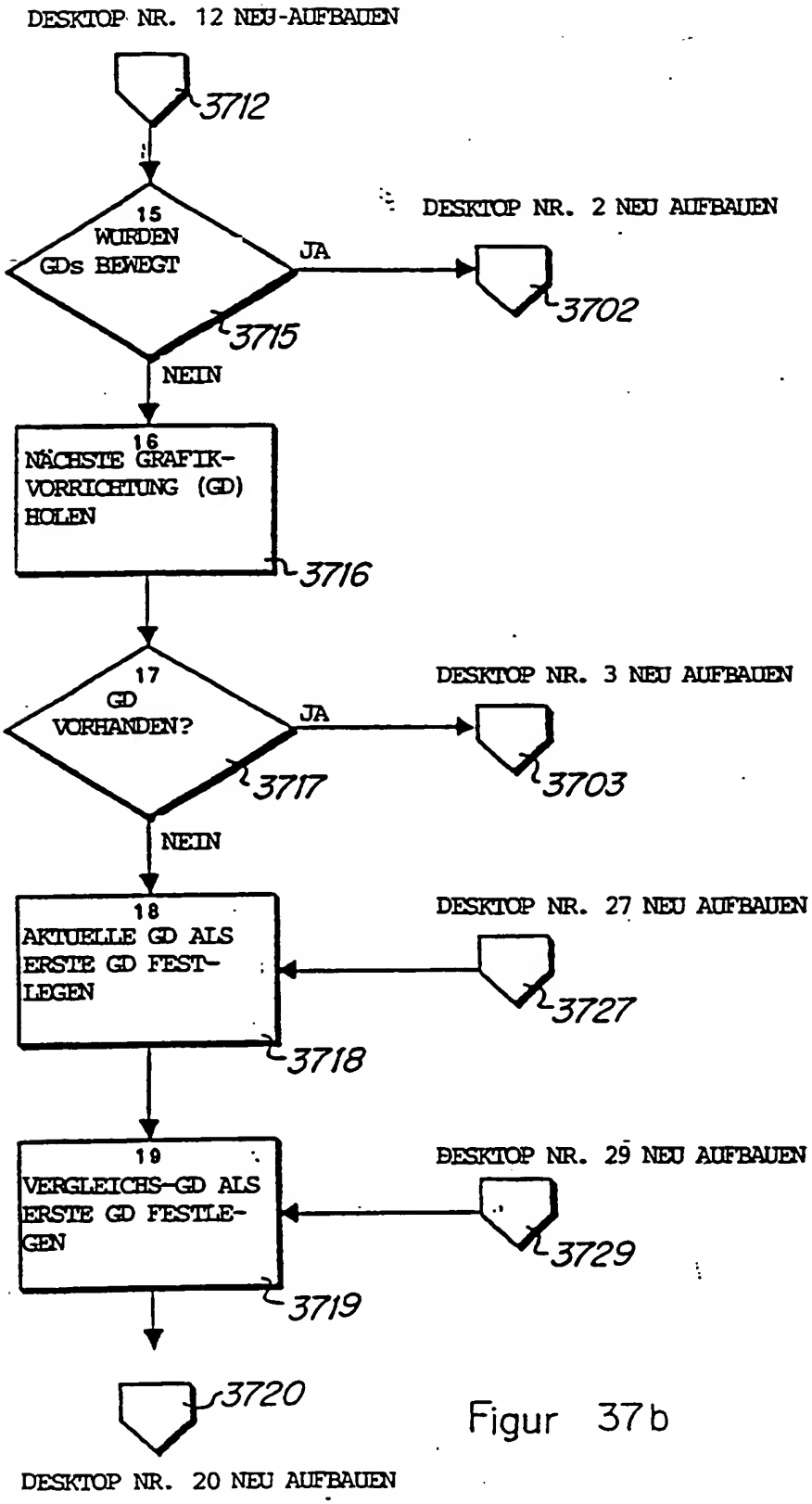
Figur 36


DESKTOP NR. 15 NEU AUFBAUEN
1
REBUILD DESKTOP
3701
2
ERSTE GRAFIKVOR-
RICHTUNG (GD)
HOLEN
3702
DESKTOP NR. 17 NEU AUFBAUEN
3
HAT SIE
EINEN OBEREN
NACHBARN ?
3703
JA
4
HAT DREHUNG
GDs GETRENNT?
3704
JA
5
OBEREN NACHBAR NACH
UNTEN BEWEGEN, SO
DASS SICH GDs
BERÜHREN
3705
NEIN
NEIN
ZU
Figur 37(a2)
ZU
Figur 37(a2)

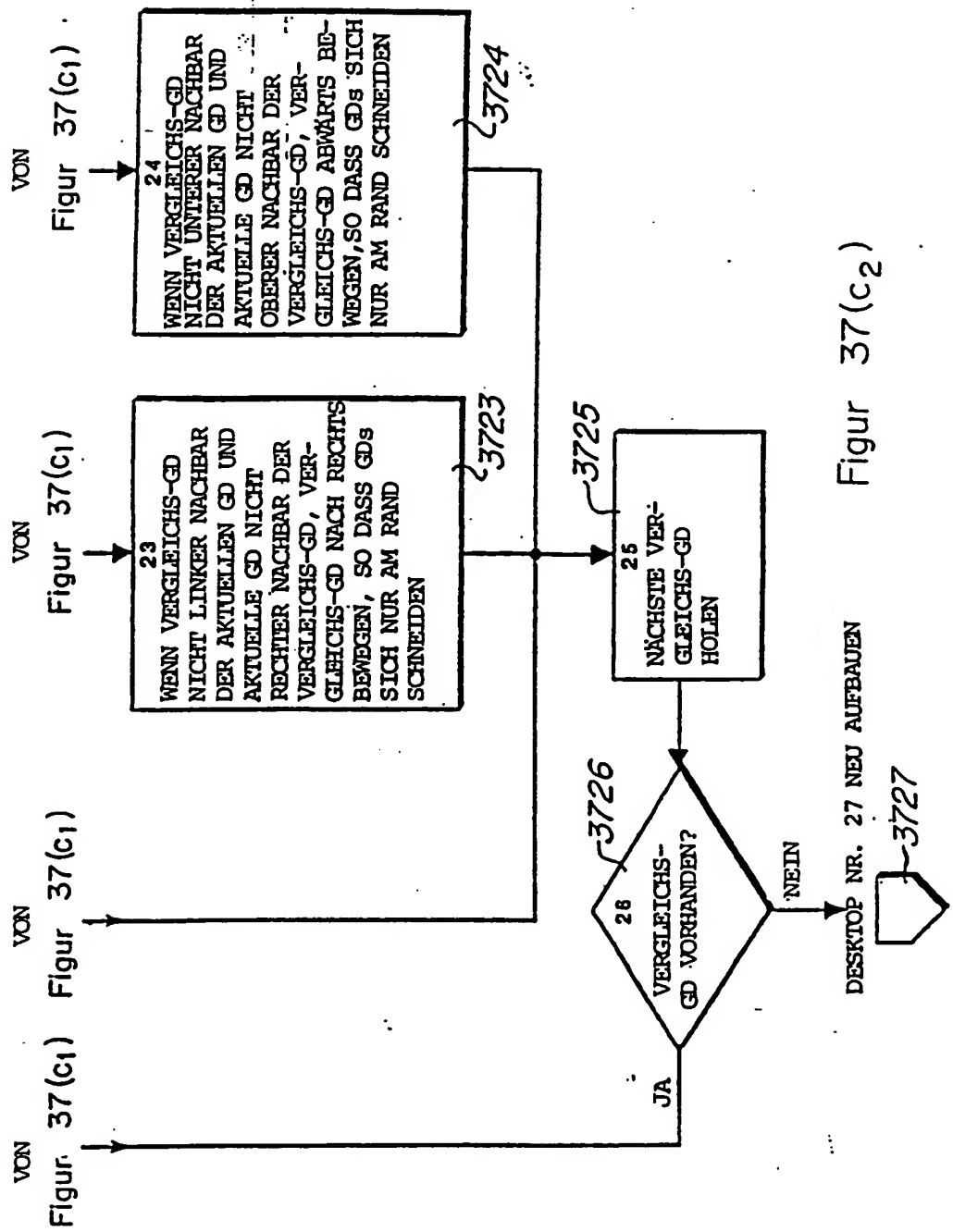
Figur 37(a1)

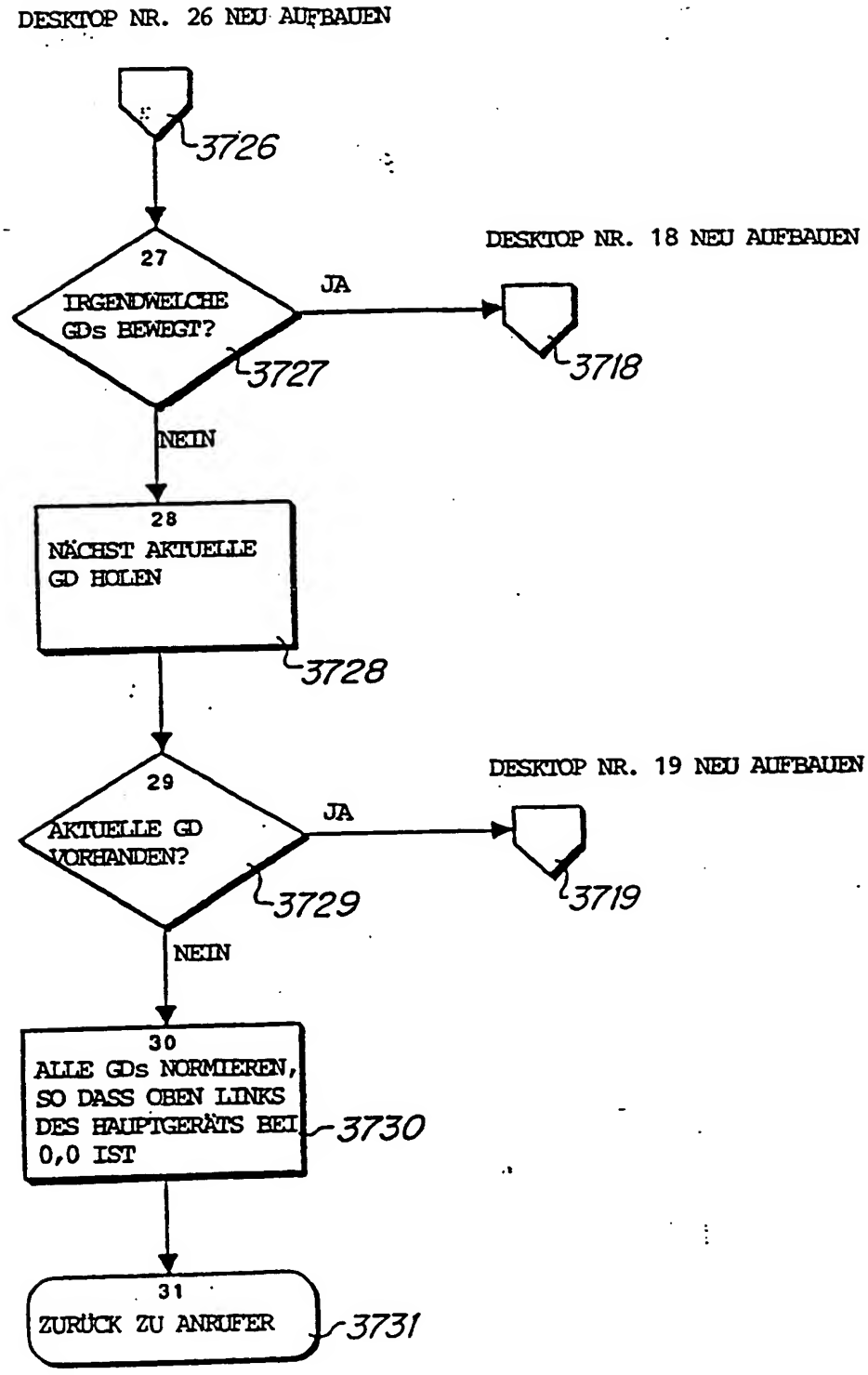

VON
Figur 37(a1)
6
HAT SIE LINKEN NACHBARN?
3706
JA
7
HAT DREHUNG GDs GETRENNT ?
3707
JA
8
LINKEN NACHBARN NACH RECHTS BEWEGEN, SO DASS SICH GDs BERÜHREN
3708
VON
Figur 37(a1)
9
HAT SIE UNTEREN NACHBARN?
3709
JA
10
HAT DREHUNG GDs GETRENNT ?
3710
JA
11
UNTEREN NACHBARN NACH OBEN BEWEGEN, SO DASS SICH GDs BERÜHREN
3711
12
HAT SIE RECHTEN NACHBARN ?
3712
JA
13
HAT DREHUNG GDs GETRENNT?
3713
JA
14
RECHTEN NACHBARN NACH LINKS BEWEGEN, SO DASS SICH GDs BERÜHREN
3714
DESKTOP NR. 15 NEU AUFBAUEN

Figur 37(a2)

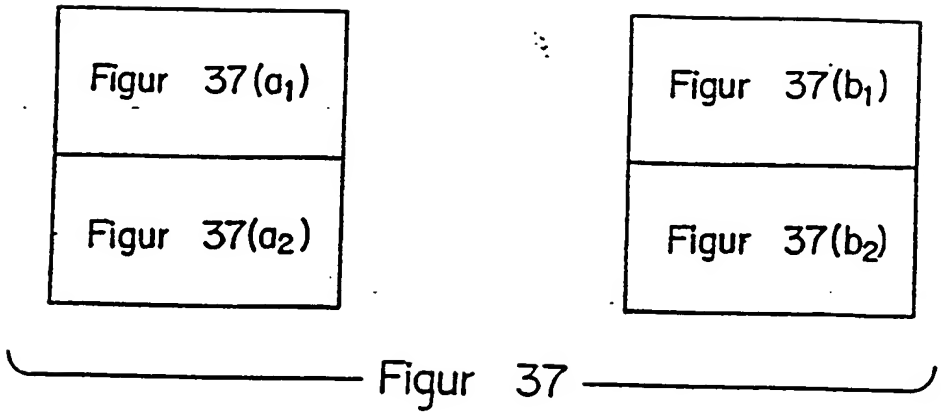

DESKTOP NR. 12 NEU-AUFBAUEN
3712
15
WURDEN GDs BEWEGT
JA
DESKTOP NR. 2 NEU AUFBAUEN
3702
3715
NEIN
16
NÄCHSTE GRAFIK-VORRICHTUNG (GD) HOLEN
3716
17
GD VORHANDEN?
JA
DESKTOP NR. 3 NEU AUFBAUEN
3703
3717
NEIN
18
AKTUELLE GD ALS ERSTE GD FEST-LEGEN
DESKTOP NR. 27 NEU AUFBAUEN
3727
3718
19
VERGLEICHS-GD ALS ERSTE GD FESTLE-GEN
DESKTOP NR. 29 NEU AUFBAUEN
3729
3719
3720
DESKTOP NR. 20 NEU AUFBAUEN

Figur 37b

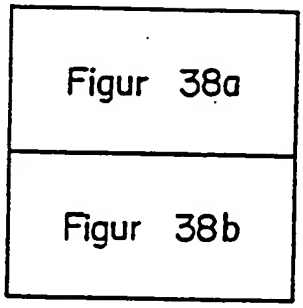

VON Figur 37(c1)
24 WENN VERGLEICHS-GD NICHT UNTERER NACHBAR DER AKTUELLEN GD UND AKTUELLE GD NICHT OBERER NACHBAR DER VERGLEICHS-GD, VERGLEICHS-GD ABWÄRTS BEWEGEN, SO DASS GDs SICH NUR AM RAND SCHNEIDEN
3724
VON Figur 37(c1)
23 WENN VERGLEICHS-GD NICHT LINKER NACHBAR DER AKTUELLEN GD UND AKTUELLE GD NICHT RECHTER NACHBAR DER VERGLEICHS-GD, VERGLEICHS-GD NACH RECHTS BEWEGEN, SO DASS GDs SICH NUR AM RAND SCHNEIDEN
3723
3725
25 NÄCHSTE VERGLEICHS-GD HOLEN
VON Figur 37(c1)
3726
26 VERGLEICHS-GD VORHANDEN?
JA
NEIN
DESKTOP NR. 27 NEU AUFBAUEN
3727
VON Figur 37(c1)
Figur 37(c2)


DESKTOP NR. 26 NEU AUFBAUEN
3726
27
IRGENDWELCHE GDs BEWEGT?
3727
JA
DESKTOP NR. 18 NEU AUFBAUEN
3718
NEIN
28
NÄCHST AKTUELLE GD HOLEN
3728
29
AKTUELLE GD VORHANDEN?
3729
JA
DESKTOP NR. 19 NEU AUFBAUEN
3719
NEIN
30
ALLE GDs NORMIEREN, SO DASS OBEN LINKS DES HAUPTGERÄTS BEI 0,0 IST
3730
31
ZURÜCK ZU ANRUFER
3731

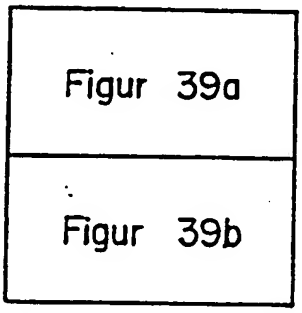
Figur 37d



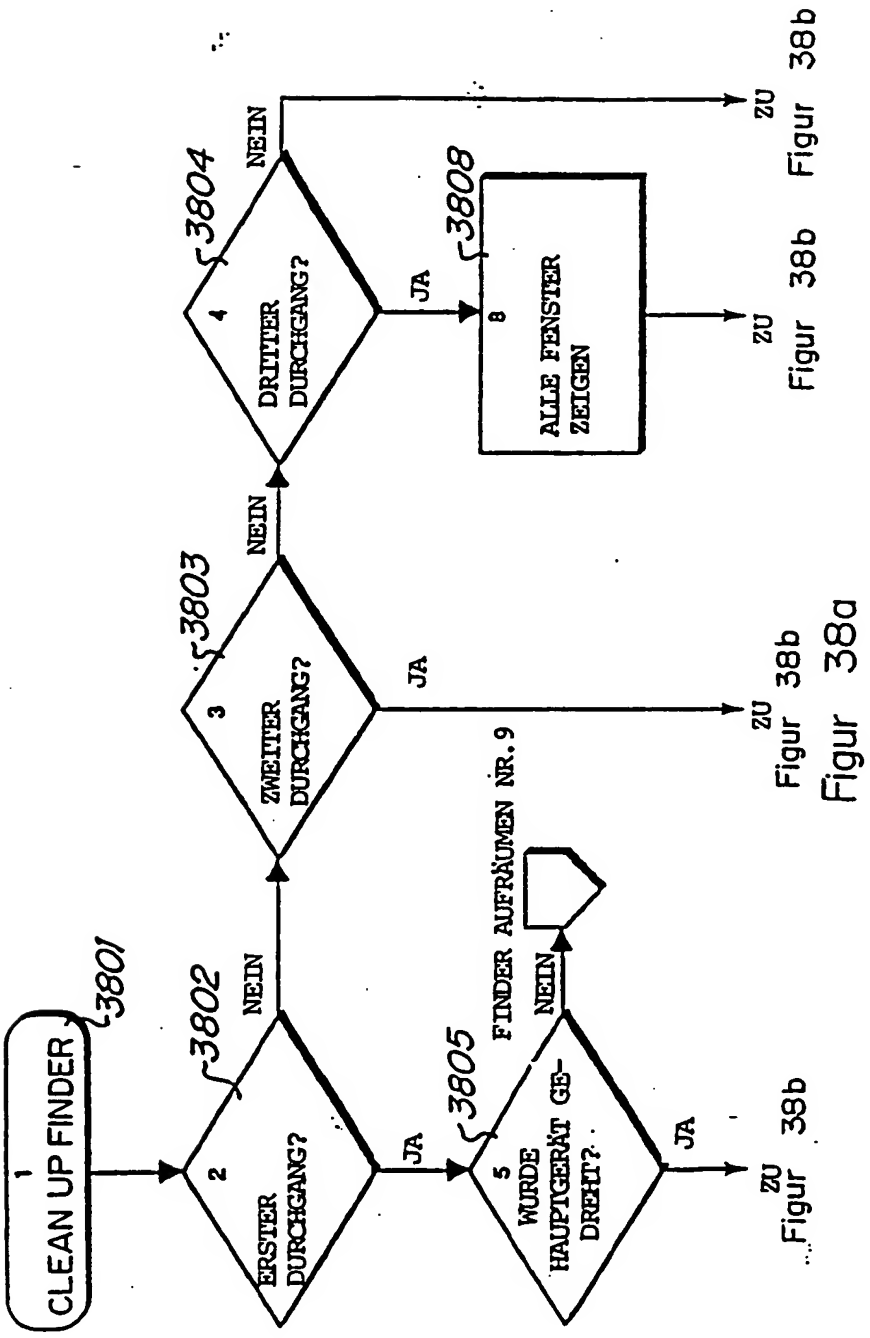
Figur 37



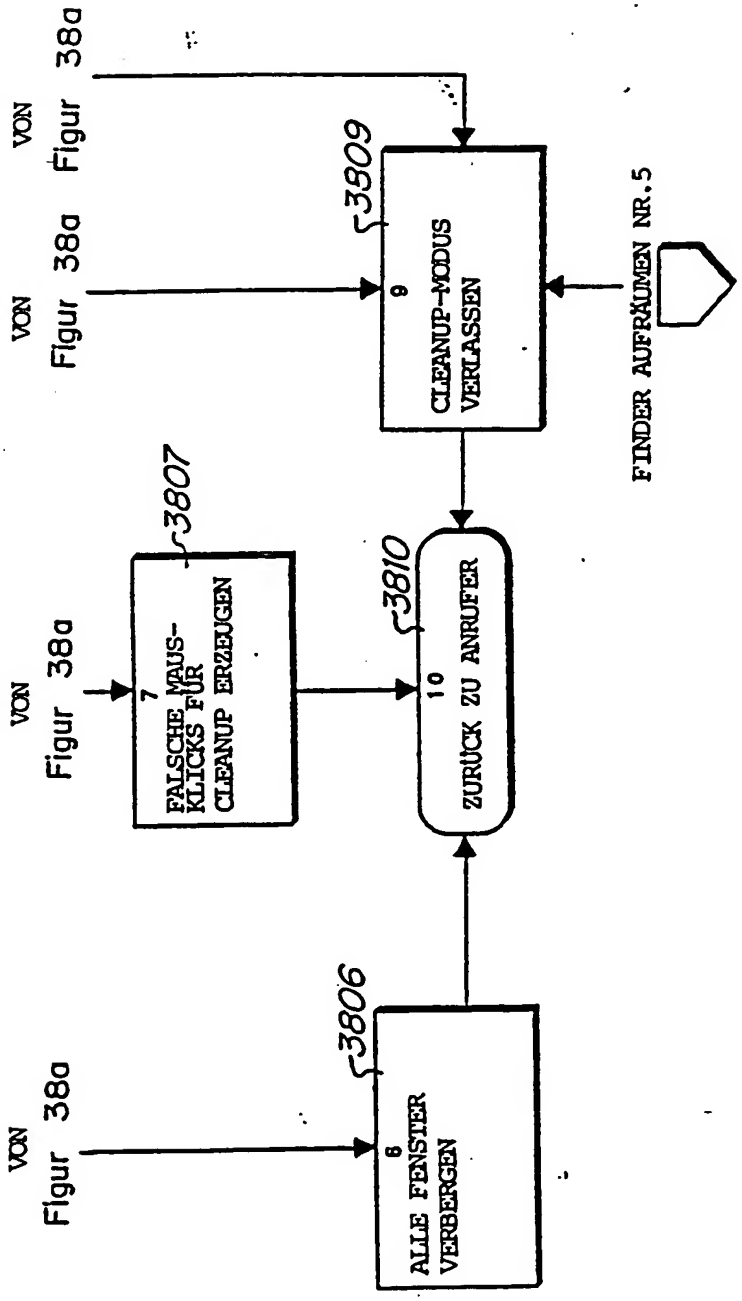
Figur 38



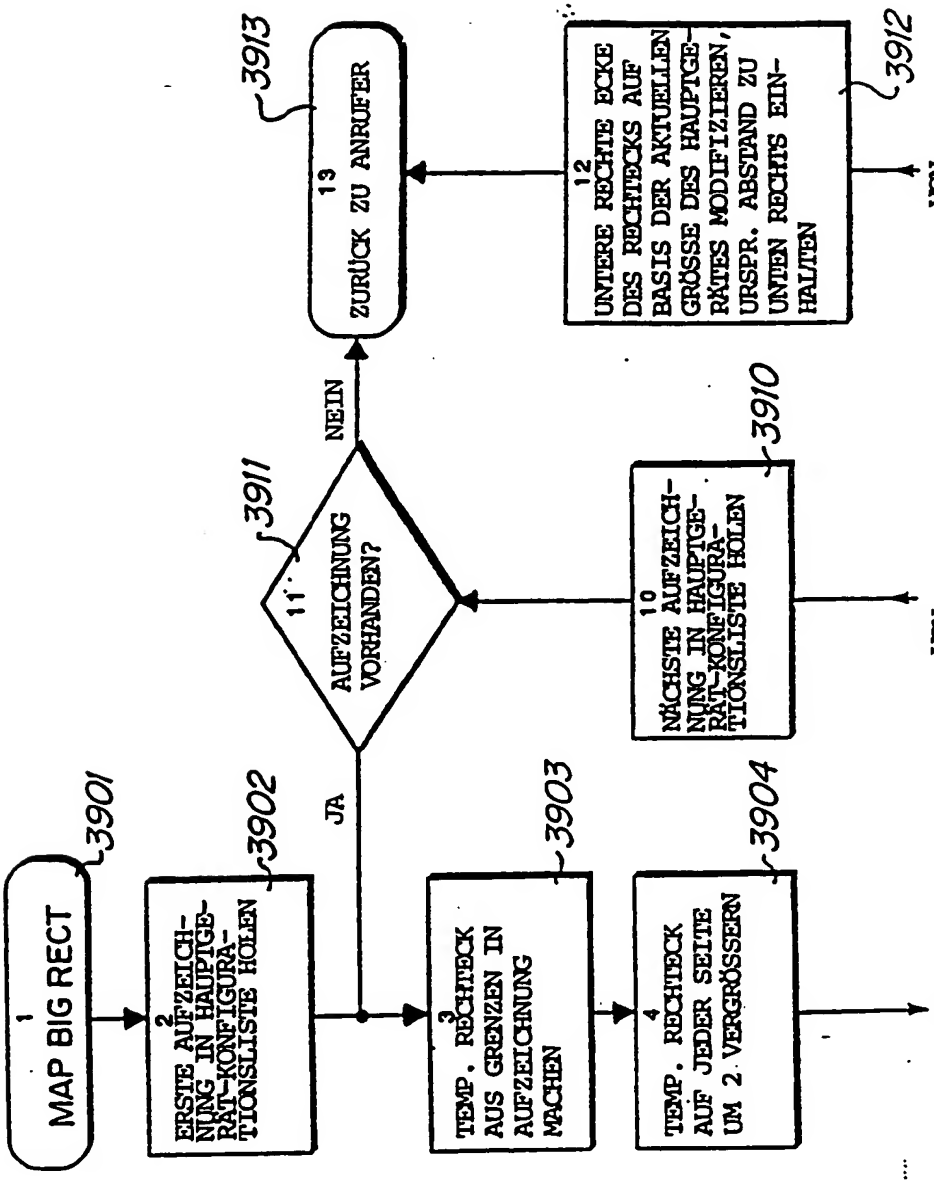
Figur 39


1
CLEAN UP FINDER
3801
2
ERSTER DURCHGANG?
3802
NEIN
JA
3
ZWEITER DURCHGANG?
3803
NEIN
JA
4
DRITTER DURCHGANG?
3804
NEIN
JA
5
WURDE HAUPTGERÄT GE-DREHT?
3805
NEIN
JA
FINDER AUFRÄUMEN NR. 9
8
ALLE FENSTER ZEIGEN
3808
ZU Figur 38b
ZU Figur 38b
ZU Figur 38b
ZU Figur 38b
ZU Figur 38b

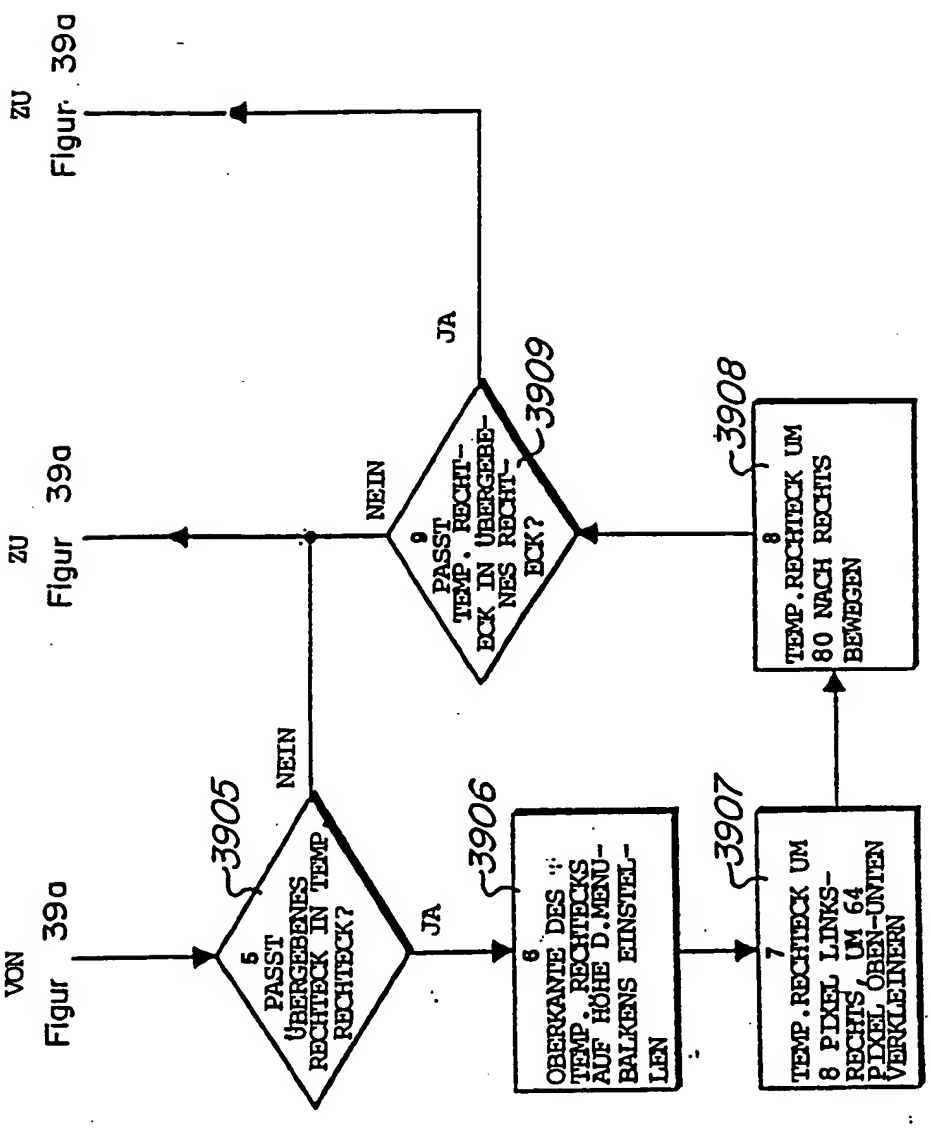
Figur 38a


VON Figur 38a
VON Figur 38a
VON Figur 38a
VON Figur 38a
3809
9
CLEANUP-MODUS VERLASSEN
FINDER AUFRÄUMEN NR.5
3807
7
FALSCHE MAUS-KLICKS FÜR CLEANUP ERZEUGEN
3810
10
ZURÜCK ZU ANRUFER
3806
6
ALLE FENSTER VERBERGEN

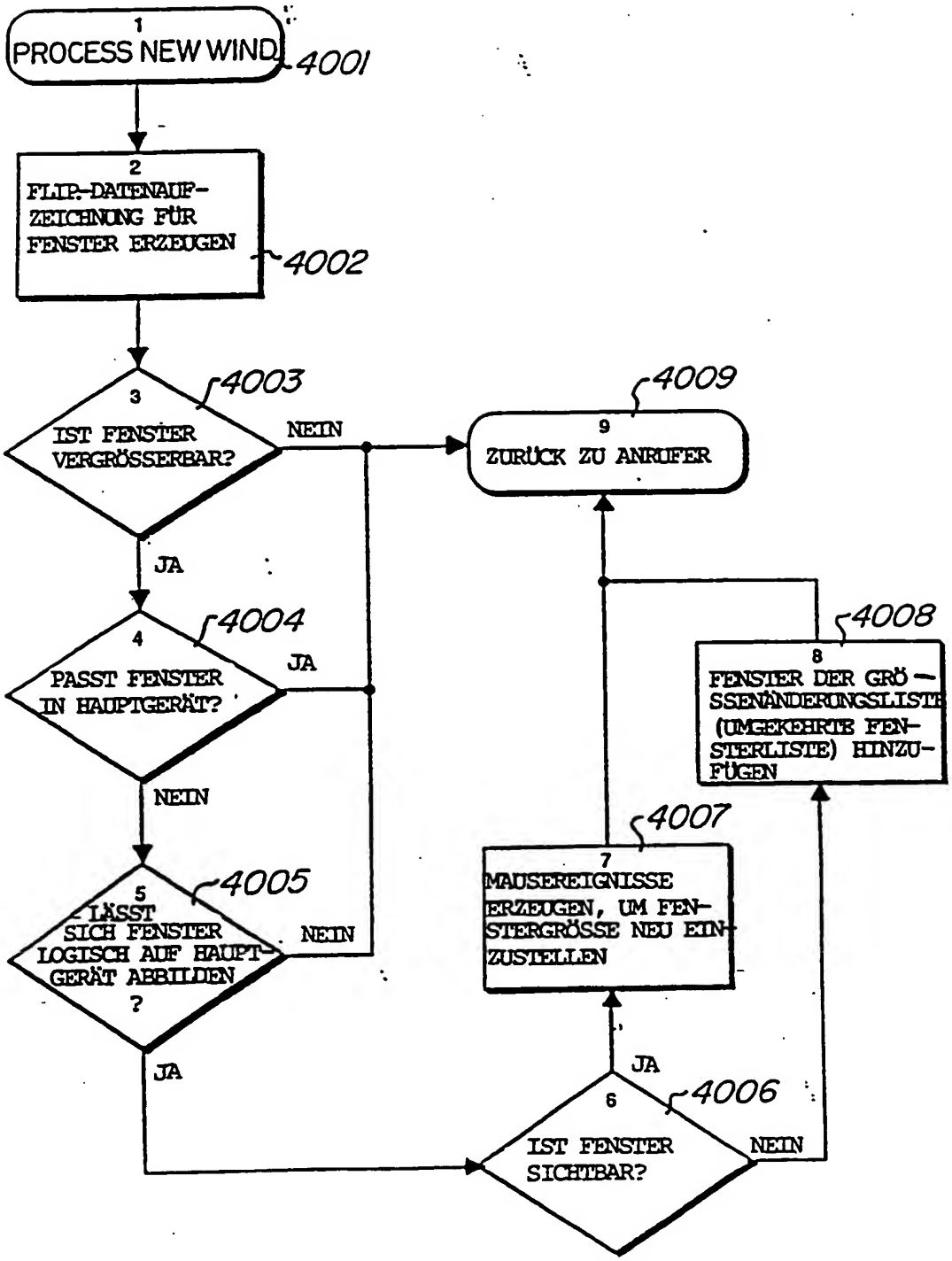
Figur 38b


1
MAP BIG RECT
3901
2
ERSTE AUFZEICHNUNG IN HAUPTGERÄT-KONFIGURATIONSLISTE HOLEN
3902
3
TEMP. RECHTECK AUS GRENZEN IN AUFZEICHNUNG MACHEN
3903
4
TEMP. RECHTECK AUF JEDER SEITE UM 2 VERGRÖSSERN
3904
ZU Figur 39b
JA
11
AUFZEICHNUNG VORHANDEN?
3911
NEIN
13
ZURÜCK ZU ANRUFER
3913
10
NÄCHSTE AUFZEICHNUNG IN HAUPTGERÄT-KONFIGURATIONSLISTE HOLEN
3910
VON Figur 39b
12
UNTERE RECHTE ECKE DES RECHTECKS AUF BASIS DER AKTUELLEN GRÖSSE DES HAUPTGERÄTES MODIFIZIEREN, URSPR. ABSTAND ZU UNTEN RECHTS EINHALTEN
3912
VON Figur 39b

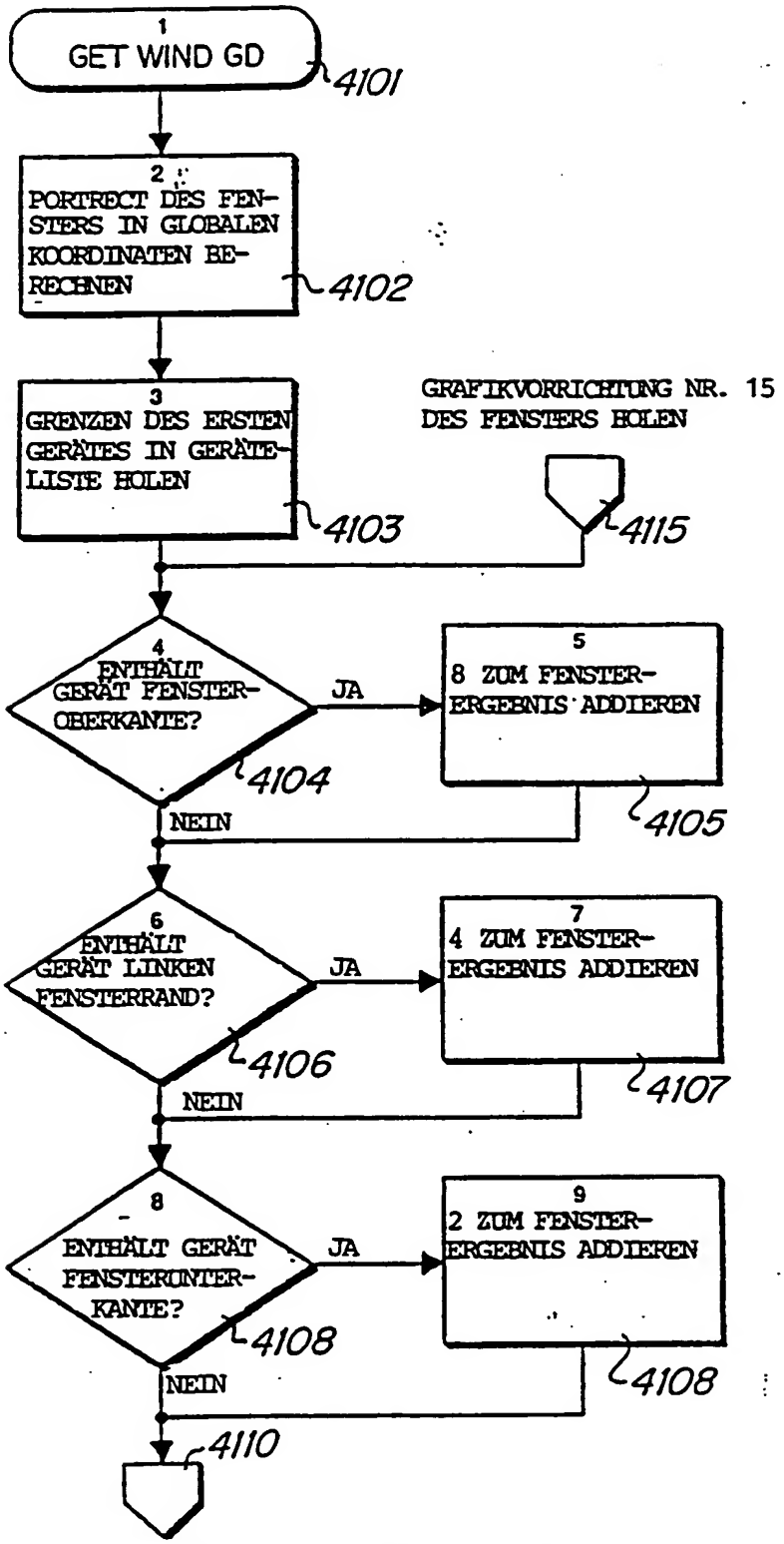
Figur 39a


VON
Figur 39a
5
PASST ÜBERGEBENES RECHTECK IN TEMP RECHTECK?
3905
NEIN
JA
6
OBERKANTE DES TEMP. RECHTECKS AUF HÖHE D. MENÜ-BALKENS EINSTELLEN
3906
7
TEMP. RECHTECK UM 8 PIXEL LINKS-RECHTS, UM 64 PIXEL OBEN-UNTEN VERKLEINERN
3907
8
TEMP. RECHTECK UM 80 NACH RECHTS BEWEGEN
3908
9
PASST TEMP. RECHTECK IN ÜBERGEBENES RECHTECK?
3909
NEIN
JA
ZU
Figur 39a
ZU
Figur 39a

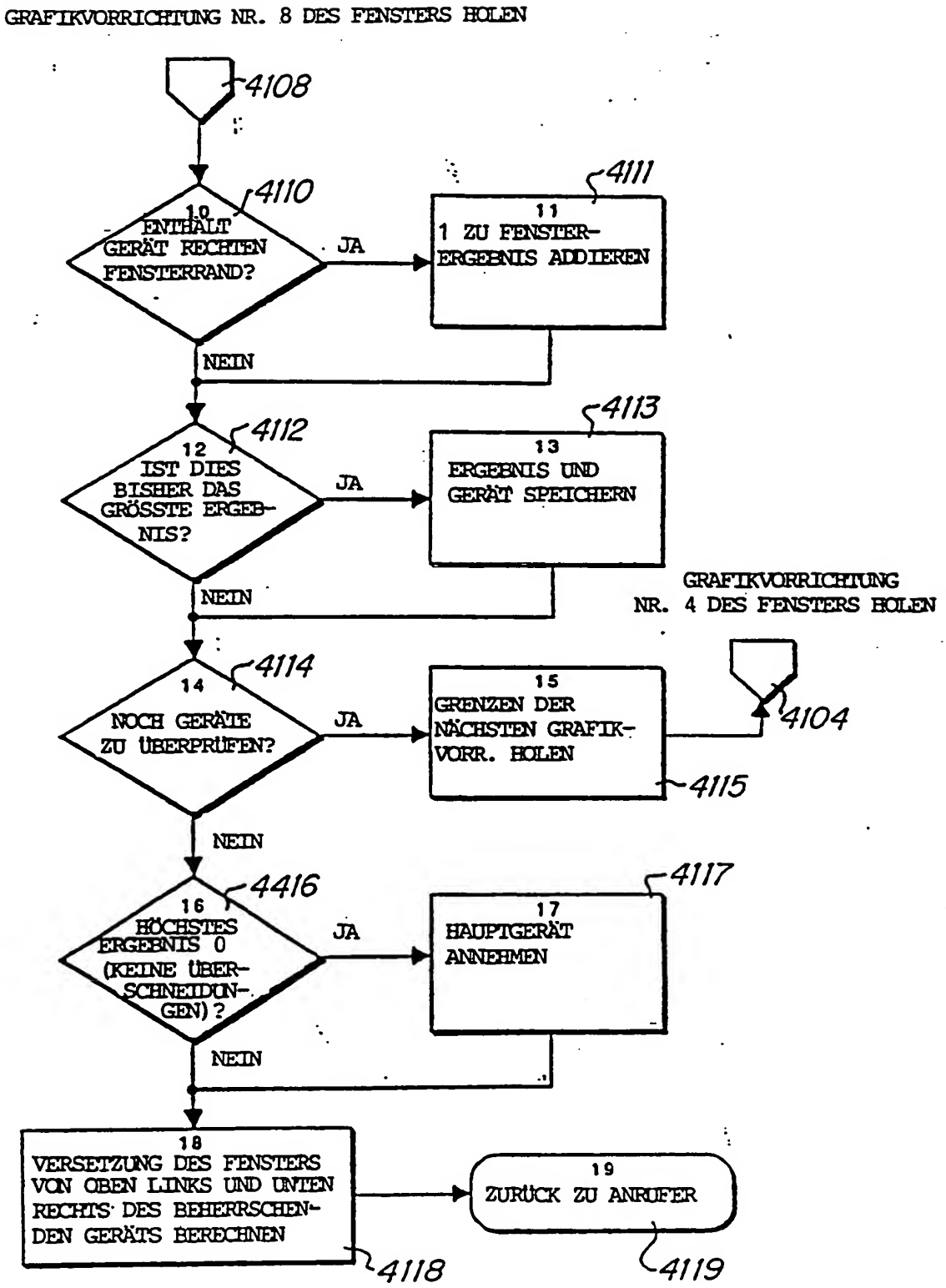
Figur 39b


1
PROCESS NEW WIND
4001
2
FLIP-DATENAUF-ZEICHNUNG FÜR FENSTER ERZEUGEN
4002
3
IST FENSTER VERGRÖSSERBAR?
4003
NEIN
JA
4
PASST FENSTER IN HAUPTGERÄT?
4004
JA
NEIN
5
LÄSST SICH FENSTER LOGISCH AUF HAUPT-GERÄT ABBILDEN ?
4005
NEIN
JA
6
IST FENSTER SICHTBAR?
4006
JA
NEIN
7
MAUSEREIGNISSE ERZEUGEN, UM FENSTERGRÖSSE NEU EINZUSTELLEN
4007
8
FENSTER DER GRÖSSENÄNDERUNGSLISTE (UMGEKEHRTE FENSTERLISTE) HINZUFÜGEN
4008
9
ZURÜCK ZU ANRUFER
4009

Figur 40


1
GET WIND GD
4101
2
PORTRECT DES FENSTERS IN GLOBALEN KOORDINATEN BERECHNEN
4102
3
GRENZEN DES ERSTEN GERÄTES IN GERÄTELISTE HOLEN
4103
GRAFIKVORRICHTUNG NR. 15 DES FENSTERS HOLEN
4115
4
ENTHÄLT GERÄT FENSTEROBERKANTE?
JA
4104
5
8 ZUM FENSTERERGEBNIS ADDIEREN
4105
NEIN
6
ENTHÄLT GERÄT LINKEN FENSTERRAND?
JA
4106
7
4 ZUM FENSTERERGEBNIS ADDIEREN
4107
NEIN
8
ENTHÄLT GERÄT FENSTERUNTERKANTE?
JA
4108
9
2 ZUM FENSTERERGEBNIS ADDIEREN
4108
NEIN
4110
GRAFIKVORRICHTUNG NR. 10 DES FENSTERS HOLEN

Figur 41a


GRAFIKVORRICHTUNG NR. 8 DES FENSTERS HOLEN
4108
4110
10
ENTHÄLT GERÄT RECHTEN FENSTERRAND?
JA
4111
11
1 ZU FENSTER-ERGEBNIS ADDIEREN
NEIN
4112
12
IST DIES BISHER DAS GRÖSSTE ERGEBNIS?
JA
4113
13
ERGEBNIS UND GERÄT SPEICHERN
NEIN
GRAFIKVORRICHTUNG NR. 4 DES FENSTERS HOLEN
4114
14
NOCH GERÄTE ZU ÜBERPRÜFEN?
JA
15
GRENZEN DER NÄCHSTEN GRAFIK-VORR. HOLEN
4104
4115
NEIN
4416
16
HÖCHSTES ERGEBNIS 0 (KEINE ÜBER-SCHNEIDUN-GEN)?
JA
4117
17
HAUPTGERÄT ANNEHMEN
NEIN
18
VERSETZUNG DES FENSTERS VON OBEN LINKS UND UNTEN RECHTS DES BEHERRSCHEN-DEN GERÄTS BERECHNEN
4118
19
ZURÜCK ZU ANRUFER
4119

Figur 41b